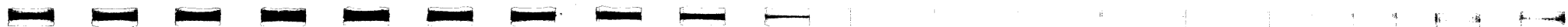


PLMs

mainp.plm
cmac3.plm
mnem1.plm
mnem4.plm
subr1.plm
scan.plm
ermod.plm
expr.plm
pseud2.plm
dline.plm
cm2.plm

cmac1.plm
cmac4.plm
mnem2.plm
symb.plm
subr2.plm
print.plm
text.plm
bexpr.plm
cmsubr.plm
global.plm

cmac2.plm
cmac5.plm
mnem3.plm
io.plm
files.plm
predef.plm
outp.plm
pseud1.plm
instr.plm
cm.plm



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE MAINP
 OBJECT MODULE PLACED IN MAINP.OBJ

COMPILER INVOKED BY: :F0: MAINP.PL M DEBJG OPTIMIZE(2) DATE(10/5/81) PAGewidth(100) XREF

```

1      $title ("ASM86 MAIN PROGRAM")
      mainp:
      do:
      /*

          This is the main program of the CP/M 8086
          assembler. This module activates the i/o
          modules and goes through the source text
          in 3 passes. The module then for each source
          line calls the external subroutine DeCODELINE
          to perform assembly of each line;

      */

      /*

          modified 3/25/81 R. Silberstein
          modified 3/28/81 R. Silberstein
          modified 3/30/81 R. Silberstein
          modified 4/7/81 R. Silberstein
          modified 4/20/81 R. Silberstein
          modified 6/16/81 R. Silberstein
          modified 7/24/81 R. Silberstein
          modified 7/27/81 R. Silberstein
          modified 8/21/81 R. Silberstein

      */

      $include (:f1:macro.lit)
=      $nolist
      $include (:f1:struc.lit)
=      $nolist
      $include (:f1:equals.lit)
=      $nolist
      $include (:f1:dev.lit)
=      $nolist
      $include (:f1:ermod.lit)
=      $nolist
      $include (:f1:subr2.ext)
=      $nolist
      $include (:f1:io.ext)
=      $nolist
      $include (:f1:files.ext)
=      $nolist
      $include (:f1:outp.ext)
=      $nolist
      $include (:f1:scan.ext)
=      $nolist
      $include (:f1:print.ext)
=      $nolist
      $include (:f1:symb.ext)
=      $nolist

```

CCP/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

```

    $include (:fil:ermod.ext)
=   $nolist
    $include (:fil:aglob.ext)
=   $nolist
    $include (:fil:text.ext)
=   $nolist
    $include (:fil:dline.ext)
=   $nolist

218 1  DECLARE ASM86 LABEL PUBLIC;

219 1  closefiles: procedure;
220 2      call close$source;
221 2      call close$print;
222 2      call close$symbol;
223 2      call close$hex;
224 2  end closefiles;

225 1  open$output$files: procedure;
226 2      CALL OPENPRINT;
227 2      CALL OPENHEX;
228 2      CALL OPENSYMBOL;
229 2  end open$output$files;

230 1  userbreak: proc byte; /* test if keyboard break from user */
231 2      if not constat then return false; /* test console status */
233 2      if readconsole then; /* skip first break key */
235 2          do forever;
236 3              call outtext(.usbreaktext); /* USER BREAK. OK (Y/N)? */
237 3              i:=upper(readconsole);
238 3              call crlf;
239 3              if i = yes then return true;
241 3              if i = no then return false;
          end$forever;
244 2  end userbreak;

245 1  varinit: proc;
246 2      CALL FILL (0, .PRINT$ON-.CIP, .CIP);
247 2      errors=0;
248 2      printswitchoff,includeon=false;
249 2      IFLEVEL = 0;
250 2      IFLIST = TRUE;
251 2      csegtype=rsc;
252 2  end varinit;

253 1  pass0init: proc; /* initialize pass 0 */
254 2      simform=false;
255 2      maxcol=119;
256 2      if printdevice=console then maxcol=79;
258 2      call symbinit; /* initialize symbol table */
259 2      print$on=false;
260 2      call varinit;
261 2  end pass0init;

262 1  pass1init: proc; /* initialize for pass 1 */
263 2      call varinit;
264 2  end pass1init;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____


```

265 1 pass2init: proc; /* initialize for pass 2 (last pass) */
266 2     print$on=true;
267 2     call varinit;
268 2     call emitinit;
269 2 end pass2init;

270 1 pass0terminate: proc; /* terminate pass 0 */
271 2     call outtext(.pass0text); /* End of pass 0 */
272 2 end pass0terminate;

273 1 pass1terminate: proc; /* terminate pass 1 */
274 2     print$on=false; /* dummy */
275 2     call outtext(.pass1text); /* End of pass 1 */
276 2 end pass1terminate;

277 1 pass2terminate: proc; /* terminate pass 2 (last pass) */
278 2     DECLARE USEFACT BYTE;
279 2     USEFACT = (FREEPT-.MEMORY) / ((ENDOFSYMPTAB-.MEMORY) / 100 + 1);
280 2     errorprinted=false;
281 2     call emitterminate; /* terminate hex output module */
282 2     call symbterminate; /* print symbols */
283 2     if printdevice <> console then call printterminate (USEFACT);
284 2     CALL CLOSEFILES;
285 2     call outtext(.endtext); /* END OF ASSEMBLY... */
286 2     call decout(errors,.help(0)); /* print no of errors */
287 2     call outtext(.help(2));
288 2     CALL OUTTEXT (.USEFACT);
289 2     CALL DECOUT (USEFACT, .HELP(0));
290 2     CALL OUTTEXT (.HELP(0));
291 2     CALL OUTTEXT (.HELP(3));
292 2     CALL WRITECONSOLE (25H); /* % */
293 2     CALL CRLF;
294 2 end pass2terminate;

295 1 include$close: proc (flag);
296 2     declare flag byte;
297 2     if eofset and include$on then$do
298 3         call close$include;
299 3         include$on,eofset=false;
300 3     if flag then$do
301 4         call scan; /* skip EOF */
302 4         call scan; /* prepare for next source line */
303 4     end$if;
304 2 end$if;
305 2 end include$close;

307 2

308 1 ASM86:

call globalinit; /* initialize some globals */

309 1 call outtext(.asm86text); /* CP/M 8086 ASSEMBLER.... */

310 1 if not filessetup then$do
311 2     call outtext(.parerrtext); /* PARAMETER ERROR */
312 2     CALL SYSTEMRESET;
313 2 end$if;
314 2

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```
315 1 CALL OPENSOURCE;
316 1 CALL OPENDOUTPUTFILES;
317 1 pass=0ffh;
318 1 do while (pass:=pass+1) < 3;
319 2   do case pass;
320 3     call pass0init;      /* pass 0 */
321 3     call pass1init;      /* pass 1 */
322 3     call pass2init;      /* pass 2 */
323 3   end$case;
324 2   call scaninit;
325 2   call scan;
326 2   do while not eofset;
327 3     if userbreak then$do
329 4       eofset=true;
330 4       pass=3;
331 4     else$do
333 4       errorprinted=false;
334 4       call decodeline;
335 4       call includeclose(false); /* close include file if necessary */
336 4       call scan;
337 4       call includeclose(true); /* close include file if necessary */
338 4     end$if;
339 3   end$while;
340 2   do case pass;
341 3     call pass0terminate; /* pass 0 */
342 3     call pass1terminate; /* pass 1 */
343 3     call pass2terminate; /* pass 2 */
344 3     do; end;             /* do nothing if userbreak */
346 3   end$case;
347 2 end$while;
348 1 call system$reset;
349 1 end$module mainp;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
2			ADDR LITERALLY 12 18 36 39 42 45 138 179 180 186 189 192 195 198 201 204 207 212
26	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(5) STACK=0000H
29	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(6) STACK=0000H
206	0000H	2	ASCIIPTR WORD PARAMETER 207
218	0010H		ASM86. LABEL PUBLIC 308
215	0000H	1	ASM86TEXT. BYTE ARRAY(1) EXTERNAL(106) 309
3			BASEREGBIT LITERALLY
3			BASEREGCOUNT LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
5			BYT. LITERALLY
23	0000H	1	CH BYTE PARAMETER 24
163	0000H	1	CH BYTE PARAMETER 164
102	0000H	1	CH BYTE PARAMETER 103
146	0000H	1	CH BYTE PARAMETER 147
26	0000H	1	CH BYTE PARAMETER 27
17	0000H	1	CH BYTE PARAMETER 18
29	0000H	1	CH BYTE PARAMETER 30
20	0000H	1	CH BYTE PARAMETER 21
105	0000H	1	CH BYTE PARAMETER 106
32	0000H	1	CH BYTE PARAMETER 33
108	0000H	1	CH BYTE PARAMETER 109
64	0000H	1	CHAR BYTE PARAMETER 65
61	0000H	1	CHAR BYTE PARAMETER 62
212	0000H	2	CIP. WORD EXTERNAL(83) 246
215	0000H	1	CLOSEERRTEXT BYTE ARRAY(1) EXTERNAL(112)
80	0000H		CLOSEFILE. PROCEDURE BYTE EXTERNAL(27) STACK=0000H
219	00CDH	17	CLOSEFILES PROCEDURE STACK=0004H 285
131	0000H		CLOSENEX PROCEDURE EXTERNAL(48) STACK=0000H 223
129	0000H		CLOSEINCLUDE PROCEDURE EXTERNAL(47) STACK=0000H 299
133	0000H		CLOSEPRINT PROCEDURE EXTERNAL(49) STACK=0000H 221
125	0000H		CLOSESOURCE. PROCEDURE EXTERNAL(45) STACK=0000H 220
135	0000H		CLOSESYMBOL. PROCEDURE EXTERNAL(50) STACK=0000H 222
4			CODE LITERALLY
10			CODEMACROERR LITERALLY
3			CODEMACROHEAD. LITERALLY
9			CONSOLE. LITERALLY 256 283
67	0000H		CONSTAT. PROCEDURE BYTE EXTERNAL(22) STACK=0000H 231
2			CR LITERALLY
92	0000H		CREATEFILE PROCEDURE BYTE EXTERNAL(31) STACK=0000H
100	0000H		CRLF PROCEDURE EXTERNAL(34) STACK=0000H 238 293
212	0000H	1	CSEGTTYPE BYTE EXTERNAL(84) 251
44	0000H	2	D. WORD PARAMETER 45
41	0000H	2	D. WORD PARAMETER 42
35	0000H	2	D. WORD PARAMETER 36
38	0000H	2	D. WORD PARAMETER 39
2			DCL. LITERALLY

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

216	0000H	DECODELINE	PROCEDURE EXTERNAL(117) STACK=0000H	314	
44	0000H	DECOUT	PROCEDURE EXTERNAL(11) STACK=0000H	287	290
4		DELETEDSYMB.	LITERALLY		
83	0000H	DELETEFILE	PROCEDURE EXTERNAL(28) STACK=0000H		
194	0000H	2 DEST	WORD PARAMETER 195		
137	0000H	1 DEV.	BYTE PARAMETER 138		
20	0000H	DIGIT.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H		
71	0000H	1 DISKNUMBER	BYTE PARAMETER 72		
215	0000H	1 DISKREADERRTEXT.	BYTE ARRAY(1) EXTERNAL(109)		
215	0000H	1 DISKWRITEERRTXT.	BYTE ARRAY(1) EXTERNAL(110)		
97	0000H	2 DMAADDRESS	WORD PARAMETER 98		
4		DOUBLEDEFINED.	LITERALLY		
10		DOUBLEDEFLAB	LITERALLY		
10		DOUBLEDEFSYMB.	LITERALLY		
10		DOUBLEDEFVAR	LITERALLY		
5		DWRD	LITERALLY		
177	0000H	EJECT.	PROCEDURE EXTERNAL(68) STACK=0000H		
2		ELSEDD	LITERALLY		
146	0000H	EMITCODEBYTE	PROCEDURE EXTERNAL(55) STACK=0000H		
142	0000H	EMITINIT	PROCEDURE EXTERNAL(53) STACK=0000H	268	
144	0000H	EMITTERMINATE.	PROCEDURE EXTERNAL(54) STACK=0000H	281	
161	0000H	EMPTYLINE.	PROCEDURE BYTE EXTERNAL(61) STACK=0000H		
56	0000H	2 ENDBUF	WORD EXTERNAL(17)		
2		ENDCASE.	LITERALLY		
2		ENDDD.	LITERALLY		
2		ENDFOREVER	LITERALLY 242		
2		ENDIF.	LITERALLY 305		
2		ENDMODULE.	LITERALLY		
2		ENDOFFILE.	LITERALLY		
10		ENDOFFLINEERR	LITERALLY		
180	0000H	2 ENDOFSYMBTAB	WORD EXTERNAL(70) 279		
2		ENDPROC.	LITERALLY		
215	0000H	1 ENDTEXT.	BYTE ARRAY(1) EXTERNAL(104)	286	
2		ENDWHILE	LITERALLY		
197	0000H	ENTERATTRIBUTES.	PROCEDURE EXTERNAL(77) STACK=0000H		
212	0000H	1 EOFSET	BYTE EXTERNAL(92) 297 300 326 329		
35	0000H	EQUAL.	PROCEDURE BYTE EXTERNAL(8) STACK=0000H		
209	0000H	ERRMSG	PROCEDURE EXTERNAL(81) STACK=0000H		
209	0000H	1 ERRNO.	BYTE PARAMETER 210		
4		ERROR.	LITERALLY		
212	0000H	1 ERRORPRINTED	BYTE EXTERNAL(93) 280 333		
212	0000H	2 ERRORS	WORD EXTERNAL(91) 247 287		
2		FALSE.	LITERALLY 232 242 248 254 259 274 280		
			300 333 335		
53	0000H	1 FCB.	BYTE ARRAY(1) EXTERNAL(14)		
54	0000H	1 FCB16.	BYTE ARRAY(1) EXTERNAL(15)		
83	0000H	2 FCBADDRESS	WORD PARAMETER 84		
92	0000H	2 FCBADDRESS	WORD PARAMETER 93		
89	0000H	2 FCBADDRESS	WORD PARAMETER 90		
86	0000H	2 FCBADDRESS	WORD PARAMETER 87		
80	0000H	2 FCBADDRESS	WORD PARAMETER 81		
77	0000H	2 FCBADDRESS	WORD PARAMETER 78		
74	0000H	2 FCBADDRESS	WORD PARAMETER 75		
14	0000H	2 FCBADR	WORD PARAMETER 15		
14	0000H	FILEABORT.	PROCEDURE EXTERNAL(1) STACK=0000H		
3		FILEISSTRUCTURE	LITERALLY		
3		FILEOSTRUCTURE	LITERALLY		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

140	0000H	FILESETUP.	PROCEDURE BYTE EXTERNAL(52) STACK=0000H	310
10		FILESYNTER.	LITERALLY	
17	0000H	FILL	PROCEDURE EXTERNAL(2) STACK=0000H	246
137	0000H	2 FILNAM	WORD PARAMETER 138	
137	0000H	2 FILTYP	WORD PARAMETER 133	
200	0000H	FINDCODEMACRO.	PROCEDURE BYTE EXTERNAL(78) STACK=0000H	
191	0000H	FINDSYMBOL	PROCEDURE BYTE EXTERNAL(75) STACK=0000H	
10		FIRSTITEM.	LITERALLY	
295	0004H	1 FLAG	BYTE PARAMETER AUTOMATIC 296 301	
2		FOREVER.	LITERALLY 235	
2		FORMFEED	LITERALLY	
179	0000H	2 FREEPT	WORD EXTERNAL(69) 279	
212	0000H	1 FULLSYMTAB.	BYTE EXTERNAL(94)	
47	0000H	1 FUNC	BYTE PARAMETER 48	
50	0000H	1 FUNC	BYTE PARAMETER 51	
194	0000H	GETATTRIBUTES.	PROCEDURE EXTERNAL(76) STACK=0000H	
213	0000H	GLOBALINIT	PROCEDURE EXTERNAL(99) STACK=0000H	308
212	0000H	5 HELP	BYTE ARRAY(5) EXTERNAL(97) 287 288 290	
			291	
38	0000H	HEX1OUT.	PROCEDURE EXTERNAL(9) STACK=0000H	
41	0000H	HEX2OUT.	PROCEDURE EXTERNAL(10) STACK=0000H	
212	0000H	1 I.	BYTE EXTERNAL(98) 237 239 241	
4		IDENT.	LITERALLY	
137	0000H	IFILESETUP	PROCEDURE EXTERNAL(51) STACK=0000H	
212	0000H	1 IFLEVEL.	BYTE EXTERNAL(96) 249	
212	0000H	1 IFLIST	BYTE EXTERNAL(87) 250	
10		IFPARERR	LITERALLY	
10		ILLEGALMACRO	LITERALLY	
10		ILLEGALPSEUDO.	LITERALLY	
10		ILLEXPRELEM.	LITERALLY	
10		ILLTOPER	LITERALLY	
295	0232H	44 INCLUDECLOSE	PROCEDURE STACK=0006H 335 337	
212	0000H	1 INCLUDEON.	BYTE EXTERNAL(95) 248 297 300	
3		INDEXREGBIT.	LITERALLY	
3		INDEXREGCOUNT.	LITERALLY	
50	0000H	2 INFO	WORD PARAMETER 51	
47	0000H	2 INFO	WORD PARAMETER 48	
113	0000H	ININCLUDEBYTE.	PROCEDURE BYTE EXTERNAL(39) STACK=0000H	
2		INIT	LITERALLY	
215	0000H	1 INITIALS	BYTE ARRAY(1) EXTERNAL(102)	
111	0000H	INSOURCEBYTE	PROCEDURE BYTE EXTERNAL(38) STACK=0000H	
10		INSTKERR	LITERALLY	
95	0000H	INTERROGATEDISK.	PROCEDURE BYTE EXTERNAL(32) STACK=0000H	
3		IREGBIT.	LITERALLY	
3		IREGCOUNT.	LITERALLY	
4		LAB.	LITERALLY	
10		LABOUTOF RANGE.	LITERALLY	
8		LEFTBRACKET.	LITERALLY	
23	0000H	LETTER	PROCEDURE BYTE EXTERNAL(4) STACK=0000H	
2		LF	LITERALLY	
206	0000H	1 LG	BYTE PARAMETER 207	
203	0000H	1 LG	BYTE PARAMETER 204	
200	0000H	1 LG	BYTE PARAMETER 201	
191	0000H	1 LG	BYTE PARAMETER 192	
188	0000H	1 LG	BYTE PARAMETER 189	
185	0000H	1 LG	BYTE PARAMETER 186	
2		LIT.	LITERALLY 2 3 4 5 6 7 8	

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

			9	10	
188	0000H	2	MACDEFPT	WORD PARAMETER	139
1	0002H	203	MAINP.	PROCEDURE STACK=0010H	
215	0000H	1	MAKEERRTEXT.	BYTE ARRAY(1) EXTERNAL(111)	
212	0000H	1	MAXCOL	BYTE EXTERNAL(88)	255 257
	0000H		MEMORY	BYTE ARRAY(0)	279
10			MISSIFERR.	LITERALLY	
10			MISSINGPSEUDO.	LITERALLY	
10			MISSINSTR.	LITERALLY	
10			MISSSEGMINFD	LITERALLY	
10			MISSSTYPETAFD	LITERALLY	
47	0000H		MON1	PROCEDURE EXTERNAL(12) STACK=0000H	
50	0000H		MON2	PROCEDURE BYTE EXTERNAL(13) STACK=0000H	
35	0000H	1	N.	BYTE PARAMETER	36
17	0000H	1	N.	BYTE PARAMETER	18
44	0000H	2	N.	WORD PARAMETER	45
41	0000H	2	N.	WORD PARAMETER	42
38	0000H	1	N.	BYTE PARAMETER	39
4			NEGLECTED.	LITERALLY	
10			NEGLECTERR.	LITERALLY	
10			NESTEDIFERR.	LITERALLY	
10			NESTEDINCLUDEERR.	LITERALLY	
203	0000H		NEWCMBODY.	PROCEDURE BYTE EXTERNAL(79) STACK=0000H	
206	0000H		NEWCMNAME.	PROCEDURE BYTE EXTERNAL(80) STACK=0000H	
188	0000H		NEWMACRO	PROCEDURE BYTE EXTERNAL(74) STACK=0000H	
185	0000H		NEWSYMBOL.	PROCEDURE BYTE EXTERNAL(73) STACK=0000H	
5			NIL.	LITERALLY	
215	0000H	1	NO	BYTE EXTERNAL(101)	241
3			NOOVCOUNT.	LITERALLY	
3			NOOVERRIDEBIT.	LITERALLY	
9			NULL	LITERALLY	
4			NUMBER	LITERALLY	
8			OAND	LITERALLY	
8			DEQ.	LITERALLY	
8			OGE.	LITERALLY	
8			OGT.	LITERALLY	
8			OLAST.	LITERALLY	
8			OLE.	LITERALLY	
8			OLENGTH.	LITERALLY	
8			OLT.	LITERALLY	
8			OMOD	LITERALLY	
8			ONE.	LITERALLY	
8			ONOT	LITERALLY	
8			OOFFSET.	LITERALLY	
8			OOR.	LITERALLY	
215	0000H	1	OPENERRTXT.	BYTE ARRAY(1) EXTERNAL(108)	
74	0000H		OPENFILE	PROCEDURE BYTE EXTERNAL(25) STACK=0000H	
119	0000H		OPENHFX.	PROCEDURE EXTERNAL(42) STACK=0000H	227
117	0000H		OPENINCLUDE.	PROCEDURE EXTERNAL(41) STACK=0000H	
225	0000H	14	OPENOUTPUTFILES.	PROCEDURE STACK=0004H	316
121	0000H		OPENPRINT.	PROCEDURE EXTERNAL(43) STACK=0000H	226
77	0000H		OPENROFILE	PROCEDURE BYTE EXTERNAL(26) STACK=0000H	
115	0000H		OPENSOURCE	PROCEDURE EXTERNAL(40) STACK=0000H	315
123	0000H		OPENSYMBOL	PROCEDURE EXTERNAL(44) STACK=0000H	228
3			OPERANDSTRUC	LITERALLY	
4			OPERATOR	LITERALLY	
10			OPMISMATCH	LITERALLY	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

8		OPTR	LITERALLY	
8		OSEG	LITERALLY	
8		OSHL	LITERALLY	
8		OSHORT	LITERALLY	
8		OSHR	LITERALLY	
8		OTYPE	LITERALLY	
102	0000H	OUTEXBYTE	PROCEDURE EXTERNAL(35) STACK=0000H	
105	0000H	OUTPRINTBYTE	PROCEDURE EXTERNAL(36) STACK=0000H	
108	0000H	OUTSYMBOLBYTE	PROCEDURE EXTERNAL(37) STACK=0000H	
11	0000H	OUTTEXT	PROCEDURE EXTERNAL(40) STACK=0000H	236 271
			275 286 289 291 309 312	
8		OXOR	LITERALLY	
215	0000H	1 PAGETEXT	BYTE ARRAY(1) EXTERNAL(103)	
215	0000H	1 PAREKRTXT	BYTE ARRAY(1) EXTERNAL(107)	312
212	0000H	1 PASS	BYTE EXTERNAL(82)	317 318 319 330 340
253	015AH	38 PASS0INIT	PROCEDURE STACK=000EH	320
270	0198H	12 PASS0TERMINATE	PROCEDURE STACK=0006H	341
215	0000H	1 PASS0TEXT	BYTE ARRAY(1) EXTERNAL(115)	271
262	0180H	8 PASS1INIT	PROCEDURE STACK=000EH	321
273	01A4H	17 PASS1TERMINATE	PROCEDURE STACK=0006H	342
215	0000H	1 PASS1TEXT	BYTE ARRAY(1) EXTERNAL(116)	275
265	0188H	16 PASS2INIT	PROCEDURE STACK=000EH	322
277	01B5H	125 PASS2TERMINATE	PROCEDURE STACK=0008H	343
7		PCODEMACRO	LITERALLY	
7		PCSEG	LITERALLY	
7		PDB	LITERALLY	
7		PDBIT	LITERALLY	
7		PDD	LITERALLY	
7		PDSEG	LITERALLY	
7		PDW	LITERALLY	
7		PEJECT	LITERALLY	
7		PEND	LITERALLY	
7		PENDIF	LITERALLY	
7		PENDM	LITERALLY	
7		PEQU	LITERALLY	
7		PESEG	LITERALLY	
7		PTF	LITERALLY	
7		PTFLIST	LITERALLY	
7		PINCLUDE	LITERALLY	
7		PLIST	LITERALLY	
7		PMDRM	LITERALLY	
7		PNOIFLIST	LITERALLY	
7		PNOLIST	LITERALLY	
7		PNOSEGFIX	LITERALLY	
7		PORG	LITERALLY	
7		PPAGESIZE	LITERALLY	
7		PPAGEWIDTH	LITERALLY	
7		PRB	LITERALLY	
7		PRELB	LITERALLY	
7		PRELW	LITERALLY	
166	0000H	PRINTCRLF	PROCEDURE EXTERNAL(63) STACK=0000H	
212	0000H	1 PRINTDEVICE	BYTE EXTERNAL(89)	256 283
9		PRINTER	LITERALLY	
172	0000H	PRINTINIT	PROCEDURE EXTERNAL(66) STACK=0000H	
168	0000H	PRINTNEWPAGE	PROCEDURE EXTERNAL(64) STACK=0000H	
212	0000H	1 PRINTON	BYTE EXTERNAL(85)	246 259 266 274
163	0000H	PRINTSINGLLBYTE	PROCEDURE EXTERNAL(62) STACK=0000H	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

170	0000H		PRINTSOURCELINE. . .	PROCEDURE EXTERNAL(65) STACK=0000H	
212	0000H	1	PRINTSWITCHOFF. . .	BYTE EXTERNAL(36) 248	
174	0000H		PRINTINTERMEDIATE. . .	PROCEDURE EXTERNAL(67) STACK=0000H	234
2			PROC.	LITERALLY 26 29 38 41 44 102 105	
				108 111 113 115 117 119 121 123 125 127	
				129 131 133 135 137 140 142 144 146 149	
				151 153 155 158 161 163 166 168 170 172	
				174 177 181 183 185 188 191 194 197 200	
				203 206 209 216 230 245 253 262 265 270	
				273 277 295	
7			PRS.	LITERALLY	
7			PRW.	LITERALLY	
7			PSEGFIX.	LITERALLY	
4			PSEUDO.	LITERALLY	
10			PSEUDOOPER.	LITERALLY	
7			PSIFORM.	LITERALLY	
7			PSSEG.	LITERALLY	
17	0000H	2	PT.	WORD PARAMETER 18	
7			PTITLE.	LITERALLY	
203	0000H	2	PTR.	WORD PARAMETER 204	
6			RAH.	LITERALLY	
6			RAL.	LITERALLY	
6			RAX.	LITERALLY	
6			RBH.	LITERALLY	
6			RBL.	LITERALLY	
6			RBP.	LITERALLY	
6			RBX.	LITERALLY	
6			RCH.	LITERALLY	
6			RCL.	LITERALLY	
6			RCS.	LITERALLY 251	
6			RCX.	LITERALLY	
6			RDH.	LITERALLY	
6			RDI.	LITERALLY	
6			RDL.	LITERALLY	
6			RDS.	LITERALLY	
6			RDX.	LITERALLY	
59	0000H		READCONSOLE. . . .	PROCEDURE BYTE EXTERNAL(19) STACK=0000H 233	
				237	
86	0000H		READRECORD. . . .	PROCEDURE BYTE EXTERNAL(29) STACK=0000H	
2			REENT.	LITERALLY	
4			REG.	LITERALLY	
6			RES.	LITERALLY	
200	0000H	2	RESULT.	WORD PARAMETER 201	
191	0000H	2	RESULT.	WORD PARAMETER 192	
185	0000H	2	RESULT.	WORD PARAMETER 186	
206	0000H	2	RETURNPTR.	WORD PARAMETER 207	
127	0000H		REWINDSOURCE. . . .	PROCEDURE EXTERNAL(46) STACK=0000H	
8			RIGHTBRACKET. . . .	LITERALLY	
6			RSI.	LITERALLY	
6			RSP.	LITERALLY	
6			RSS.	LITERALLY	
35	0000H	2	S.	WORD PARAMETER 36	
151	0000H		SCAN.	PROCEDURE EXTERNAL(57) STACK=0000H 303 304	
				325 336	
149	0000H		SCANINIT.	PROCEDURE EXTERNAL(56) STACK=0000H 324	
3			SEGM8IT.	LITERALLY	
3			SEGMCOUNT.	LITERALLY	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

3		SEGTYPEBIT	LITERALLY	
3		SEGTYPECOUNT	LITERALLY	
71	0000H	SELECTDISK	PROCEDURE EXTERNAL(24) STACK=0000H	
97	0000H	SETDMAADDRESS	PROCEDURE EXTERNAL(33) STACK=0000H	
212	0000H	1 SIMFORM	BYTE EXTERNAL(10) 254	
153	0000H	SKIPRESTOFLINE	PROCEDURE EXTERNAL(58) STACK=0000H	
158	0000H	SKIPUNTIL	PROCEDURE BYTE EXTERNAL(60) STACK=0000H	
197	0000H	2 SOURCE	WORD PARAMETER 198	
2		SPACE	LITERALLY	
4		SPEC	LITERALLY	
155	0000H	SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(59) STACK=0000H	
200	0000H	2 STRADR	WORD PARAMETER 201	
191	0000H	2 STRADR	WORD PARAMETER 192	
188	0000H	2 STRADR	WORD PARAMETER 189	
185	0000H	2 STRADR	WORD PARAMETER 186	
4		STRING	LITERALLY	
2		STRUC	LITERALLY	
197	0000H	2 SYMBADR	WORD PARAMETER 198	
194	0000H	2 SYMBADR	WORD PARAMETER 195	
215	0000H	1 SYMBFULLERRTEXT	BYTE ARRAY(1) EXTERNAL(113)	
183	0000H	SYMBINIT	PROCEDURE EXTERNAL(72) STACK=0000H	258
4		SYMBOL	LITERALLY	
3		SYMBOLHEAD	LITERALLY	
3		SYMBOLSTRUC	LITERALLY	
181	0000H	SYMBTERMINATE	PROCEDURE EXTERNAL(71) STACK=0000H	282
57	0000H	SYSTEMRESET	PROCEDURE EXTERNAL(18) STACK=0000H	313 348
11	0000H	2 T	WORD PARAMETER 12	
2		TAB	LITERALLY	
55	0000H	1 TBUFF	BYTE ARRAY(1) EXTERNAL(16)	
14	0000H	2 TEXTADR	WORD PARAMETER 15	
2		THENDO	LITERALLY 297 301 310 327	
158	0000H	1 TOK	BYTE PARAMETER 159	
155	0000H	1 TOK	BYTE PARAMETER 156	
2		TRUE	LITERALLY 235 240 250 266 329 337	
146	0000H	1 TYP	BYTE PARAMETER 147	
3		TYPEBIT	LITERALLY	
3		TYPECOUNT	LITERALLY	
4		UDEFSYMB	LITERALLY	
10		UDEFSYMBOL	LITERALLY	
32	0000H	UPPER	PROCEDURE BYTE EXTERNAL(7) STACK=0000H	237
215	0000H	1 USBREAKTEXT	BYTE ARRAY(1) EXTERNAL(114)	236
278	0000H	1 USEFACT	BYTE 279 284 290	
174	0000H	1 USEFACT	BYTE PARAMETER 175	
215	0000H	1 USEFACTOR	BYTE ARRAY(1) EXTERNAL(105)	283
230	0000H	63 USERBREAK	PROCEDURE BYTE STACK=0000H	327
9		VALIDDISK	LITERALLY	
4		VARIABLE	LITERALLY	
245	0120H	47 VARINIT	PROCEDURE STACK=0000H 260 263 267	
69	0000H	VERSION	PROCEDURE WORD EXTERNAL(23) STACK=0000H	
5		WRD	LITERALLY	
61	0000H	WRITECONSOLE	PROCEDURE EXTERNAL(20) STACK=0000H	292
64	0000H	WRITELST	PROCEDURE EXTERNAL(21) STACK=0000H	
89	0000H	WRITERECORD	PROCEDURE BYTE EXTERNAL(30) STACK=0000H	
215	0000H	1 YES	BYTE EXTERNAL(100) 239	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

MODULE INFORMATION:

CODE AREA SIZE	= 025FH	606D
CONSTANT AREA SIZE	= 0000H	00
VARIABLE AREA SIZE	= 0001H	1D
MAXIMUM STACK SIZE	= 0010H	16D
932 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

ASIS-11 PL/M-86 V2.0 COMPILATION OF MODULE CMAC1

OBJECT MODULE PLACED IN CMAC1.OBJ

COMPILER INVOKED BY: SF0: CMAC1.PLM D1BJG OPTIMIZE(2) DATE(10/5/81) PAGewidth(100)

\$title ("CODEMACRO DEFINITIONS - PART 1")

cmac1:

do;

\$include (:f1:macro.lit)

= \$nolist

\$include (:f1:cmacd.lit)

= \$nolist

\$include (:f1:equals.lit)

= \$nolist

\$include (:f1:cmac.lit)

= \$nolist

/* Code-macro table: */

18 1 dcl

/* AAA */

aaal cmac3struc public data(

nil,0,

mdbn,37h,

/* DB 37H */

mendm),

/* ENDM */

/* AAD */

aadl cmac4struc public data(

nil,0,

mdwn,0d5h,0ah,

/* DW 0AD5H */

mendm),

/* ENDM */

/* AAM */

aaml cmac4struc public data(

nil,0,

mdwn,0d4h,0ah,

/* DW 0AD4H */

mendm),

/* ENDM */

/* AAS */

aasl cmac3struc public data(

nil,0,

mdbn,3fh,

/* DB 3FH */

mendm),

/* ENDM */

/* ADC dst:Eb,src:Db */

adcl cmac14struc data(

nil,2,

specE,modb,

specD,modb,

msegfix,dst,

/* SEGFIX dst */

mdbn,80h,

/* DB 80H */

mmodrm1,2,dst,

/* MODRM 2,dst */

mdbf,src,

/* DB src */

mendm),

/* ENDM */

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

```

/* ADC dst:Ew,src:Dsb */
adc2 cmac14struc data(
    .adc1,2,
    specE,modw,
    specD,modb,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,81h,         /* D3 81H */
    mmodrm1,2,dst,    /* MODRM 2,dst */
    mdwf,src,         /* DW src */
    mendm),           /* ENDM */

```

```

/* ADC dst:Ew,src:Dsb */
adc3 cmac14struc data(
    .adc2,2,
    specE,modw,
    specD,modsb,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,83h,         /* D8 83H */
    mmodrm1,2,dst,    /* MODRM 2,dst */
    mdbf,src,         /* D8 src */
    mendm),           /* ENDM */

```

```

/* ADC dst:Ew,src:Dw */
adc4 cmac14struc data(
    .adc3,2,
    specE,modw,
    specD,modw,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,81h,         /* D8 81H */
    mmodrm1,2,dst,    /* MODRM 2,dst */
    mdwf,src,         /* DW src */
    mendm),           /* ENDM */

```

```

/* ADC dst:Ab,src:Dsb */
adc5 cmac9struc data(
    .adc4,2,
    specA,modb,
    specD,modb,
    mdbn,14h,         /* D8 14H */
    mdbf,src,         /* D8 src */
    mendm),           /* ENDM */

```

```

/* ADC dst:Aw,src:Dsb */
adc6 cmac9struc data(
    .adc5,2,
    specA,modw,
    specD,modb,
    mdbn,15h,         /* D8 15H */
    mdwf,src,         /* DW src */
    mendm),           /* ENDM */

```

```

/* ADC dst:Aw,src:Dw */
adc7 cmac9struc data(
    .adc6,2,
    specA,modw,
    specD,modw,

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mdbn,15h,          /* DB 15H */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* ADC dst:Eb,src:Rb */
adc8    cmac12struc data(
        .adc7,2,
        specE,modb,
        specR,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,10h,          /* DB 10H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* ADC dst:Ew,src:Rw */
adc9    cmac12struc data(
        .adc8,2,
        specE,modw,
        specR,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,11h,          /* DB 11H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* ADC dst:Rb,src:Eb */
adc10   cmac12struc data(
        .adc9,2,
        specR,modb,
        specE,modb,
        msegfix,src,       /* SEGFIX src */
        mdbn,12h,          /* DB 12H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

/* ADC dst:Rw,src:Ew */
adc11   cmac12struc public data(
        .adc10,2,
        specR,modw,
        specE,modw,
        msegfix,src,       /* SEGFIX src */
        mdbn,13h,          /* DB 13H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

/* ADD dst:Eb,src:Db */
add1    cmac14struc data(
        nil,2,
        specE,modb,
        specD,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,80h,          /* DB 80H */
        mmodrm1,0,dst,     /* MODRM 0,dst */
        mdbf,src,          /* DB src */
        mendum),          /* ENDM */

/* ADD dst:Ew,src:Db */
add2    cmac14struc data(

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

      .addl,2,
      specE,modw,
      specD,modb,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,81h,         /* DB 81H */
      mmodrm1,0,dst,    /* MODRM 0,dst */
      mdwf,src,         /* DW src */
      mendum),         /* ENDM */

/* ADD dst:Ew,src:Dsb */
add3  cmac14struc data(
      .add2,2,
      specE,modw,
      specD,modsb,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,83h,         /* DB 83H */
      mmodrm1,0,dst,    /* MODRM 0,dst */
      mdbf,src,         /* DB src */
      mendum),         /* ENDM */

/* ADD dst:Ew,src:Dw */
add4  cmac14struc data(
      .add3,2,
      specE,modw,
      specD,modw,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,81h,         /* DB 81H */
      mmodrm1,0,dst,    /* MODRM 0,dst */
      mdwf,src,         /* DW src */
      mendum),         /* ENDM */

/* ADD dst:Ab,src:Db */
add5  cmac9struc data(
      .add4,2,
      specA,modb,
      specD,modb,
      mdbn,04h,         /* DB 04H */
      mdbf,src,         /* DB src */
      mendum),         /* ENDM */

/* ADD dst:Aw,src:Db */
add6  cmac9struc data(
      .add5,2,
      specA,modw,
      specD,modb,
      mdbn,05h,         /* DB 05H */
      mdwf,src,         /* DW src */
      mendum),         /* ENDM */

/* ADD dst:Aw,src:Dw */
add7  cmac9struc data(
      .add6,2,
      specA,modw,
      specD,modw,
      mdbn,05h,         /* DB 05H */
      mdwf,src,         /* DW src */
      mendum),         /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

add8      /* ADD dst:Eb,src:Rb */
          cmac12struc data(
            .add7,2,
            specE,modb,
            specR,modb,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,0,          /* DB 00H */
            mmodrm2,src,dst,  /* MODRM src,dst */
            mendum),        /* ENDM */

add9      /* ADD dst:Ew,src:Rw */
          cmac12struc data(
            .add8,2,
            specE,modw,
            specR,modw,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,1,          /* DB 01H */
            mmodrm2,src,dst,  /* MODRM src,dst */
            mendum),        /* ENDM */

add10     /* ADD dst:Rb,src:Eb */
          cmac12struc data(
            .add9,2,
            specR,modb,
            specE,modb,
            msegfix,src,      /* SEGFIX src */
            mdbn,2,          /* DB 02H */
            mmodrm2,dst,src,  /* MODRM dst,src */
            mendum),        /* ENDM */

add11     /* ADD dst:Rw,src:Ew */
          cmac12struc public data(
            .add10,2,
            specR,modw,
            specE,modw,
            msegfix,src,      /* SEGFIX src */
            mdbn,3,          /* DB 03H */
            mmodrm2,dst,src,  /* MODRM dst,src */
            mendum),        /* ENDM */

and1      /* AND dst:Eb,src:Db */
          cmac14struc data(
            nil,2,
            specE,modb,
            specD,modb,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,80h,         /* DB 80H */
            mmodrm1,4,dst,    /* MODRM 4,dst */
            mdbf,src,         /* DB src */
            mendum),        /* ENDM */

and2      /* AND dst:Ew,src:Db */
          cmac14struc data(
            .and1,2,
            specE,modw,specD,modb,
            msegfix,dst,      /* SEGFIX dst */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mdbn,81h,          /* DB 81H */
        mmodrm1,4,dst,     /* MODRM 4,dst */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* AND dst:Ew,src:)w */
and3    cmac14struc data(
        .and2,2,
        specE,modw,specD,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,81h,          /* DB 81H */
        mmodrm1,4,dst,     /* MODRM 4,dst */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* AND dst:Ab,src:)b */
and4    cmac9struc data(
        .and3,2,
        specA,modb,specD,modb,
        mdbn,24h,          /* DB 24H */
        mdwf,src,          /* DB src */
        mendum),          /* ENDM */

/* AND dst:Aw,src:)b */
and5    cmac9struc data(
        .and4,2,
        specA,modw,specD,modb,
        mdbn,25h,          /* DB 25H */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* AND dst:Aw,src:)w */
and6    cmac9struc data(
        .and5,2,
        specA,modw,specD,modw,
        mdbn,25h,          /* DB 25H */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* AND dst:Eb,src:)b */
and7    cmac12struc data(
        .and6,2,
        specE,modb,specR,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,20h,          /* DB 20H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* AND dst:Ew,src:)w */
and8    cmac12struc data(
        .and7,2,
        specE,modw,specR,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,21h,          /* DB 21H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____


```

and9      /* AND dst:R0,src:Eb */
          cmac12struc data(
            .and8,2,
            specR,modb,specE,modb,
            msegfix,src,      /* SEGFIX src */
            mdbn,22h,        /* DB 22H */
            mmodrm2,dst,src,  /* MODRM dst,src */
            mendm),          /* ENDM */

and10     /* AND dst:R0,src:Ew */
          cmac12struc public data(
            .and9,2,
            specR,modw,specE,modw,
            msegfix,src,      /* SEGFIX src */
            mdbn,23h,        /* DB 23H */
            mmodrm2,dst,src,  /* MODRM dst,src */
            mendm),          /* ENDM */

call1     /* CALL adr:Ew */
          cmac10struc data(
            nil,1,
            specE,modw,
            msegfix,adr,      /* SEGFIX adr */
            mdbn,0ffh,        /* DB 0FFH */
            mmodrm1,2,adr,    /* MODRM 2,adr */
            mendm),          /* ENDM */

call2     /* CALL adr:Cb */
          cmac7struc data(
            .call1,1,
            specC,modb,
            mdbn,0e8h,        /* DB 0E8H */
            mrelw,adr,        /* RELW adr */
            mendm),          /* ENDM */

call3     /* CALL adr:Cw */
          cmac7struc public data(
            .call2,1,
            specC,modw,
            mdbn,0e8h,        /* DB 0E8H */
            mrelw,adr,        /* RELW adr */
            mendm),          /* ENDM */

callf1    /* CALLF adr:Ed */
          cmac10struc data(
            nil,1,
            specE,modd,
            msegfix,adr,      /* SEGFIX adr */
            mdbn,0ffh,        /* DB 0FFH */
            mmodrm1,3,adr,    /* MODRM 3,adr */
            mendm),          /* ENDM */

callf2    /* CALLF adr:Cd */
          cmac7struc public data(
            .callf1,1,
            specC,modd,
            mdbn,9ah,        /* DB 9AH */

```

CCP/M-86 2-0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER # CC-104

```

      addf,adr,          /* DD adr */
      mendum),          /* ENDM */

/* CBW */
cbw1  cmac3struc public data(
      nil,0,
      mdbn,98h,          /* DB 98H */
      mendum),          /* ENDM */

/* CLC */
clc1  cmac3struc public data(
      nil,0,
      mdbn,0f8h,         /* DB 0F8H */
      mendum),          /* ENDM */

/* CLD */
cld1  cmac3struc public data(
      nil,0,
      mdbn,0fch,         /* DB 0FCH */
      mendum),          /* ENDM */

/* CLI */
cli1  cmac3struc public data(
      nil,0,
      mdbn,0fah,         /* DB 0FAH */
      mendum),          /* ENDM */

/* CMC */
cmc1  cmac3struc public data(
      nil,0,
      mdbn,0f5h,         /* DB 0F5H */
      mendum),          /* ENDM */

/* CMP dst:Eb,src:Db */
cmp1  cmac14struc data(
      nil,2,
      specE,modb,specD,modb,
      msegfix,dst,        /* SEGFIX dst */
      mdbn,80h,           /* DB 80H */
      mmodrml,7,dst,      /* MODRM 7,dst */
      mdbf,src,           /* DB src */
      mendum),           /* ENDM */

/* CMP dst:Ew,src:Db */
cmp2  cmac14struc data(
      .cmp1,2,
      specE,modw,specD,modb,
      msegfix,dst,        /* SEGFIX dst */
      mdbn,81h,           /* DB 81H */
      mmodrml,7,dst,      /* MODRM 7,dst */
      mdwf,src,           /* DW src */
      mendum),           /* ENDM */

/* CMP dst:Ew,src:Dsb */
cmp3  cmac14struc data(
      .cmp2,2,
      specE,modw,specD,modsb,

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

msegfix,dst,      /* SEGFIX dst */
mdbn,83h,         /* DB 83H */
mmodrm1,7,dst,    /* MODRM 7,dst */
mddf,src,         /* DB src */
mendm),          /* ENDM */

/* CMP dst:Ew,src:Dw */
cmp4 cmac14struc data(
    .cmp3,2,
    specE,modw,specD,modw,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,81h,         /* DB 81H */
    mmodrm1,7,dst,    /* MODRM 7,dst */
    mddf,src,         /* DW src */
    mendm),          /* ENDM */

/* CMP dst:Ab,src:Db */
cmp5 cmac9struc data(
    .cmp4,2,
    specA,modb,specD,modb,
    mdbn,3ch,         /* DB 3CH */
    mddf,src,         /* DB src */
    mendm),          /* ENDM */

/* CMP dst:Aw,src:Db */
cmp6 cmac9struc data(
    .cmp5,2,
    specA,modw,specD,modb,
    mdbn,3dh,         /* DB 3DH */
    mddf,src,         /* DW src */
    mendm),          /* ENDM */

/* CMP dst:Aw,src:Dw */
cmp7 cmac9struc data(
    .cmp6,2,
    specA,modw,specD,modw,
    mdbn,3dh,         /* DB 3DH */
    mddf,src,         /* DW src */
    mendm),          /* ENDM */

/* CMP dst:Eb,src:Rb */
cmp8 cmac12struc data(
    .cmp7,2,
    specE,modb,specR,modb,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,38h,         /* DB 38H */
    mmodrm2,src,dst,  /* MODRM src,dst */
    mendm),          /* ENDM */

/* CMP dst:Ew,src:Rw */
cmp9 cmac12struc data(
    .cmp8,2,
    specE,modw,specR,modw,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,39h,         /* DB 39H */
    mmodrm2,src,dst,  /* MODRM src,dst */
    mendm),          /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

/* CMP dst:Rb,src:Lb */
cmp10  cmac12struc data(
      .cmp9,2,
      specR,modb,specE,modb,
      msegfix,src,      /* SEGFIX src */
      mdbn,3ah,        /* DB 3AH */
      mmodrm2,dst,src,  /* MODRM dst,src */
      mendm),          /* ENDM */

/* CMP dst:Rw,src:Ew */
cmp11  cmac12struc public data(
      .cmp10,2,
      specR,modw,specE,modw,
      msegfix,src,      /* SEGFIX src */
      mdbn,3bh,        /* DB 3BH */
      mmodrm2,dst,src,  /* MODRM dst,src */
      mendm);          /* ENDM */

```

19 1 end\$module cmac1;

MODULE INFORMATION:

CODE AREA SIZE	= 0000H	00
CONSTANT AREA SIZE	= 02EFH	7510
VARIABLE AREA SIZE	= 0000H	00
MAXIMUM STACK SIZE	= 0000H	00
826 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

IS45-11 PL/M-86 V2.0 COMPILATION OF MODULE CHAC2

OBJECT MODULE PLACED IN CHAC2.OBJ

COMPILER INVOKED BY: :F0: CHAC2.PL M DEBJG OPTIMIZE(2) DATE(10/5/81) PAGELWIDTH(100)

```

1      $title ("CODEMACRO DEFINITIONS - PART 2")
      cmac2:
      do:

      /*

      modified 6/16/81 R. Silberstein

      */

      $include (:f1:macro.lit)
=      $nolist
=      $include (:f1:cmacd.lit)
=      $nolist
=      $include (:f1:equals.lit)
=      $nolist
=      $include (:f1:cmac.lit)
=      $nolist

18 1    dcl

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

      /* CMPS dst:Eb,src:Eb */
      cmac12struc data(
      nil,2,
      specE,modb,specE,modb,
      mnosegfix,res,dst, /* NOSEGFIX ES,dst */
      msegfix,src, /* SEGFIX src */
      mdbn,0a6h, /* DB 0A6H */
      mendm), /* ENDM */

      /* CMPS dst:Ew,src:Ew */
      cmac12struc public data(
      .cmac1,2,
      specE,modw,specE,modw,
      mnosegfix,res,dst, /* NOSEGFIX ES,dst */
      msegfix,src, /* SEGFIX src */
      mdbn,0a7h, /* DB 0A7H */
      mendm), /* ENDM */

      /* CMPSB */
      CMAC3STRUC PUBLIC DATA(
      NIL,0,
      MDBN,0A6H,
      MENDM),

      /* CMPSW */
      CMAC3STRUC PUBLIC DATA(
      NIL,0,
      MDBN,0A7H,
      MENDM),

```

```

/* CWD */
cwl1  cmac3struc public data(
      nil,0,
      mdbn,99h,          /* DB 99H */
      mendm),           /* ENDM */

/* DAA */
daa1  cmac3struc public data(
      nil,0,
      mdbn,27h,          /* DB 27H */
      mendm),           /* ENDM */

/* DAS */
das1  cmac3struc public data(
      nil,0,
      mdbn,2fh,          /* DB 2FH */
      mendm),           /* ENDM */

/* DEC dst:Eb */
dec1  cmac10struc data(
      nil,1,
      specE,mdb,
      msegfix,dst,       /* SEGFIX dst */
      mdbn,0feh,         /* DB 0FEH */
      mmodrm1,1,dst,     /* MODRM 1,dst */
      mendm),           /* ENDM */

/* DEC dst:Ew */
dec2  cmac10struc data(
      .dec1,1,
      specE,mdbw,
      msegfix,dst,       /* SEGFIX dst */
      mdbn,0ffh,         /* DB 0FFH */
      mmodrm1,1,dst,     /* MODRM 1,dst */
      mendm),           /* ENDM */

/* DEC dst:Rw */
dec3  cmac12struc public data(
      .dec2,1,
      specR,mdbw,
      mdbit,             /* DBIT 5(9H),3(dst) */
      mnumberbits,5,9h,
      mformalbits,3,dst,0,
      mendbit,
      mendm),

/* DIV divisor:Eb */
div1  cmac10struc data(
      nil,1,
      specE,mdb,
      msegfix,divisor,   /* SEGFIX divisor */
      mdbn,0f6h,         /* DB 6FH */
      mmodrm1,6,divisor, /* MODRM divisor */
      mendm),           /* ENDM */

/* DIV divisor:Ew */
div2  cmac10struc public data(

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

        .divl,1,
        specE,modw,
        msegfix,divisor,      /* SEGFIX divisor */
        mdbn,0f7h,            /* DB 0F7H */
        mmodrm1,0,divisor,    /* MODRM 6,divisor */
        mendm),               /* ENDM */

/* ESC opcode:Db(0,63),src:Eb */
esc1    cmac21struc data(
        nil,2,
        spec0,modb+doublerrange+numberrange,0,63,
        specE,modb,
        msegfix,src,          /* SEGFIX src */
        mdbit,                /* DBIT 5(1BH),3(opcode(3)) */
        mnumberbits,5,1bh,
        mformalbits,3,opcode,3,
        mendbit,
        mmodrm2,opcode,src,    /* MODRM opcode,src */
        mendm),               /* ENDM */

/* ESC opcode:Db(0,63),src:Es */
esc2    cmac21struc data(
        .esc1,2,
        spec0,modb+doublerrange+numberrange,0,63,
        specE,modw,
        msegfix,src,          /* SEGFIX src */
        mdbit,                /* DBIT 5(1BH),3(opcode(3)) */
        mnumberbits,5,1bh,
        mformalbits,3,opcode,3,
        mendbit,
        mmodrm2,opcode,src,    /* MODRM opcode,src */
        mendm),               /* ENDM */

/* ESC opcode:Db(0,63),src:Ed */
esc3    cmac21struc public data(
        .esc2,2,
        spec0,modb+doublerrange+numberrange,0,63,
        specE,modd,
        msegfix,src,          /* SEGFIX src */
        mdbit,                /* DBIT 5(1BH),3(opcode(3)) */
        mnumberbits,5,1bh,
        mformalbits,3,opcode,3,
        mendbit,
        mmodrm2,opcode,src,    /* MODRM opcode,src */
        mendm),               /* ENDM */

/* HLT */
hlt1    cmac3struc public data(
        nil,0,
        mdbn,0f4h,            /* DB 0F4H */
        mendm),               /* ENDM */

/* IDIV divisor:Eb */
idiv1   cmac10struc data(
        nil,1,
        specE,modb,
        msegfix,divisor,      /* SEGFIX divisor */

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93350

SER. # _____

```

        mdcn,0f6h,          /* DB 0F6H */
        mmodrm1,7,divisor, /* MODRM 7,divisor */
        mendum),          /* ENDM */

/* IDIV divisor:Ew */
imdiv2 cmac10struc public data(
        .idiv1,1,
        specE,modw,
        msegfix,divisor, /* SEGFIX divisor */
        mdcn,0f7h,       /* DB 0F7H */
        mmodrm1,7,divisor, /* MODRM 7,divisor */
        mendum),

/* IMUL mplier:Eb */
imul1 cmac10struc data(
        nil,1,
        specE,modb,
        msegfix,mplier, /* SEGFIX mplier */
        mdcn,0f6h,      /* DB 0F6H */
        mmodrm1,5,mplier, /* MODRM 5,mplier */
        mendum),        /* ENDM */

/* IMUL mplier:Ew */
imul2 cmac10struc public data(
        .imul1,1,
        specE,modw,
        msegfix,mplier, /* SEGFIX mplier */
        mdcn,0f7h,      /* DB 0F7H */
        mmodrm1,5,mplier, /* MODRM 5,mplier */
        mendum),        /* ENDM */

/* IN dst:Ab,port:Db */
in1 cmac9struc data(
        nil,2,
        specA,modb,specD,modb,
        mdcn,0e4h,      /* DB 0E4H */
        mdbf,port,      /* DB port */
        mendum),        /* ENDM */

/* IN dst:Aw,port:Db */
in2 cmac9struc data(
        .in1,2,
        specA,modw,specD,modb,
        mdcn,0e5h,      /* DB 0E5H */
        mdbf,port,      /* DB port */
        mendum),        /* ENDM */

/* IN dst:Ab,port:Rw(DX) */
in3 cmac8struc data(
        .in2,2,
        specA,modb,
        specR,modw+singlerange+register$range,rdx,
        mdcn,0ech,      /* DB 0ECH */
        mendum),

/* IN dst:Aw,port:Rw(DX) */
in4 cmac8struc public data(

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

        .in3,2,
        specA,modw,
        specR,modw+singlerange+register$range,rdx,
        mdbn,0adh,          /* DB 0ADH */
        mendm),            /* ENDM */

inc1    /* INC dst:Eb */
        cmac10struc data(
        nil,1,
        specE,modb,
        msegfix,dst,        /* SEGFIX dst */
        mdbn,0feh,          /* DB 0FEH */
        mmodrm1,0,dst,      /* MODRM 0,dst */
        mendm),            /* ENDM */

inc2    /* INC dst:Ew */
        cmac10struc data(
        .inc1,1,
        specE,modw,
        msegfix,dst,        /* SEGFIX dst */
        mdbn,0ffh,          /* DB 0FFH */
        mmodrm1,0,dst,      /* MODRM 0,dst */
        mendm),            /* ENDM */

inc3    /* INC dst:Rw */
        cmac12struc public data(
        .inc2,1,
        specR,modw,
        mdbit,              /* DBIT 5(08H),3(dst(0)) */
        mnumberbits,5,08h,
        mformalbits,3,dst,0,
        mendbit,
        mendm),            /* ENDM */

int1    /* INT itype:Db */
        cmac7struc data(
        nil,1,
        specD,modb,
        mdbn,0cdh,          /* DB 0CDH */
        mdbf,itype,         /* DB itype */
        mendm),            /* ENDM */

int2    /* INT itype:Db(3) */
        cmac6struc public data(
        .int1,1,
        specD,modb+singlerange,3,
        mdbn,0cch,          /* DB 0CCH */
        mendm),            /* ENDM */

into1   /* INTO */
        cmac3struc public data(
        nil,0,
        mdbn,0ceh,          /* DB 0CEH */
        mendm),            /* ENDM */

/* TRET */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
iret1  cmac3struc public data(
        nil,0,
        mdbn,0cfh,          /* DB 0CFH */
        mendm),             /* ENDM */

/* JA place:Cb */
jal     cmac7struc public data(
        nil,1,
        specC,mdb,
        mdbn,77h,          /* DB 77h */
        mrelb,place,       /* RELB place */
        mendm),            /* ENDM */

/* JAE place:Cb */
jael    cmac7struc public data(
        nil,1,
        specC,mdb,
        mdbn,73h,          /* DB 73H */
        mrelb,place,       /* RELB place */
        mendm),            /* ENDM */

/* JB place:Cb */
jbl     cmac7struc public data(
        nil,1,
        specC,mdb,
        mdbn,72h,          /* DB 72H */
        mrelb,place,       /* RELB place */
        mendm),            /* ENDM */

/* JBE place:Cb */
jbel    cmac7struc public data(
        nil,1,
        specC,mdb,
        mdbn,76h,          /* DB 76H */
        mrelb,place,       /* RELB place */
        mendm),            /* ENDM */

/* JCXZ place:Cb */
jcxz1   cmac7struc public data(
        nil,1,
        specC,mdb,
        mdbn,0e3h,          /* DB 0E3H */
        mrelb,place,       /* RELB place */
        mendm),            /* ENDM */

/* JE place:Cb */
jel     cmac7struc public data(
        nil,1,
        specC,mdb,
        mdbn,74h,          /* DB 74H */
        mrelb,place,       /* RELB place */
        mendm),            /* ENDM */

/* JG place:Cb */
jgl     cmac7struc public data(
        nil,1,
        specC,mdb,
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mdbn,7fh,          /* DB 7FH */
        mrelb,place,      /* RELB place */
        mendm),          /* ENDM */

/* JGE place:Cb */
jge1   cmac7struc public data(
        nil,1,
        specC,modb,
        mdbn,7ch,        /* DB 7CH */
        mrelb,place,      /* RELB place */
        mendm),          /* ENDM */

/* JL place:Cb */
jl1    cmac7struc public data(
        nil,1,
        specC,modb,
        mdbn,7ch,        /* DB 7CH */
        mrelb,place,      /* RELB place */
        mendm),          /* ENDM */

/* JLE place:Cb */
jle1   cmac7struc public data(
        nil,1,
        specC,modb,
        mdbn,7eh,        /* DB 7EH */
        mrelb,place,      /* RELB place */
        mendm),          /* ENDM */

/* JMP place:Ew */
jmp1   cmac10struc data(
        nil,1,
        specE,modw,
        msegfix,place,    /* SEGFIX place */
        mdbn,0ffh,        /* DB 0FFH */
        mmodrm1,4,place,  /* MODRM 4,place */
        mendm),          /* ENDM */

/* JMP place:Cw */
jmp2   cmac7struc public data(
        .jmp1,1,
        specC,modw,
        mdbn,0e9h,        /* DB 0E9H */
        mrelw,place,      /* RELW place */
        mendm),          /* ENDM */

/* JMPF place:Md */
jmpf1  cmac10struc data(
        nil,1,
        specM,modd,
        msegfix,place,    /* SEGFIX place */
        mdbn,0ffh,        /* DB 0FFH */
        mmodrm1,5,place,  /* MODRM 5,place */
        mendm),          /* ENDA */

/* JMPF place:Cd */
jmpf2  cmac7struc public data(
        .jmpf1,1,

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

specC,modd,
mdbn,0eah,      /* DB 0EAH */
mddf,place,     /* DD place */
mendm),         /* ENDM */

/* JMPs place:Cb */
jmps1 cmac7struc public data(
nil,1,
specC,modb,
mdbn,0ebh,      /* DB 0EBH */
mrelb,place,    /* RELB place */
mendm),         /* ENDM */

/* JNE place:Cb */
jne1 cmac7struc public data(
nil,1,
specC,modb,
mdbn,75h,       /* DB 75H */
mrelb,place,    /* RELB place */
mendm),         /* ENDM */

/* JNO place:Cb */
jno1 cmac7struc public data(
nil,1,
specC,modb,
mdbn,71h,       /* DB 71H */
mrelb,place,    /* RELB place */
mendm),         /* ENDM */

/* JNP place:Cb */
jnp1 cmac7struc public data(
nil,1,
specC,modb,
mdbn,7bh,       /* DB 7BH */
mrelb,place,    /* RELB place */
mendm),         /* ENDM */

/* JNS place:Cb */
jns1 cmac7struc public data(
nil,1,
specC,modb,
mdbn,79h,       /* DB 79H */
mrelb,place,    /* RELB place */
mendm),         /* ENDM */

/* JO place:Cb */
jo1 cmac7struc public data(
nil,1,
specC,modb,
mdbn,70h,       /* DB 70H */
mrelb,place,    /* RELB place */
mendm),         /* ENDM */

/* JP place:Cb */
jpl cmac7struc public data(
nil,1,
specC,modb,

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mdbn,7ah,          /* DB 7Ah */
        mrelb,place,      /* RELB place */
        mendm),          /* ENDM */

/* JS place:Cb */
jsl     cmac7struc public data(
        nil,1,
        specC,modb,
        mdbn,78h,          /* DB 78H */
        mrelb,place,      /* RELB place */
        mendm),          /* ENDM */

/* LAHF */
lahf1   cmac3struc public data(
        nil,0,
        mdbn,9fh,          /* DB 9FH */
        mendm),          /* ENDM */

/* LDS dst:Rw,src:Ed */
lds1    cmac12struc public data(
        nil,2,
        specR,modw,specE,modd,
        msegfix,src,       /* SEGFIX src */
        mdbn,0c5h,         /* DB 0C5H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendm),          /* ENDM */

/* LES dst:Rw,src:Ed */
les1    cmac12struc public data(
        nil,2,
        specR,modw,specE,modd,
        msegfix,src,       /* SEGFIX src */
        mdbn,0c4h,         /* DB 0C4H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendm),          /* ENDM */

/* LEA dst:Rw,src:M */
leal    cmac10struc public data(
        nil,2,
        specR,modw,specM,nomod,
        mdbn,8ah,          /* DB 8AH */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendm),          /* ENDM */

/* LOCK prefix */
lock1   cmac3struc public data(
        nil,0+prefixon,
        mdbn,0f0h,         /* DB 0F0H */
        mendm),          /* ENDM */

/* LODS SI$ptr:Eb */
lodsl   cmac7struc data(
        nil,1,
        specE,modb,
        msegfix,si$ptr,    /* SEGFIX SI$ptr */
        mdbn,0ach,         /* DB 0ACH */
        mendm),          /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

/* LODS ST$ptr:Ew */
lods2  cmac7struc public data(
      .lods1,1,
      specE,modw,
      msegfix,si$ptr,    /* SEGFIX ST$ptr */
      mdun,0adh,         /* DB 0ADH */
      mendm),            /* ENDM */

```

```

/* LODSB */
LODSB1 CMAC3STRUC PUBLIC DATA(
      NIL,0,
      MDBN,0ACH,
      MENDM),

```

```

/* LODSW */
LODSW1 CMAC3STRUC PUBLIC DATA(
      NIL,0,
      MDBN,0ADH,
      MENDM),

```

```

/* LOOP place:Cb */
loop1  cmac7struc public data(
      nil,1,
      specC,modb,
      mdbn,0e2h,         /* DB 0E2H */
      mrelb,place,       /* RELB place */
      mendm),            /* ENDM */

```

```

/* LODPE place:Cb */
looppe1 cmac7struc public data(
      nil,1,
      specC,modb,
      mdbn,0e1h,         /* DB 0E1H */
      mrelb,place,       /* RELB place */
      mendm),            /* ENDM */

```

```

/* LOOPNE place:Cb */
loopne1 cmac7struc public data(
      nil,1,
      specC,modb,
      mdbn,0e0h,         /* DB 0E0H */
      mrelb,place,       /* RELB place */
      mendm);            /* ENDM */

```

```

19 1  end$module cmac2;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

MODULE INFORMATION:

```

CODE AREA SIZE      = 0000H      0D
CONSTANT AREA SIZE  = 02CEH      718D
VARIABLE AREA SIZE  = 0000H      0D
MAXIMUM STACK SIZE  = 0000H      0D
852 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

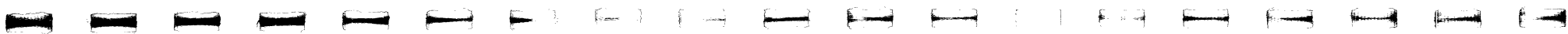
COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE CMAC3

OBJECT MODULE PLACED IN CMAC3.OBJ

COMPILER INVOKED BY: :F0: CMAC3.PLN DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100)

```

1      $title ('CODEMACRO DEFINITIONS - PART 3')
      cmac3:
      do:

      /*

      modified 6/16/81 R. Silberstein

      */

      $include (:f1:macro.lit)
=      $nolist
=      $include (:f1:cmacd.lit)
=      $nolist
=      $include (:f1:equals.lit)
=      $nolist
=      $include (:f1:cmac.lit)
=      $nolist

18 1    dcl

      /* MOV dst:Eb,src:Db */
mov1    cmac14struc data(
      nil,2,
      specE,modb,specD,modb,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0c6h,        /* DB 0C6h */
      mmodrm1,0,dst,    /* MODRM 0,dst */
      mdbf,src,         /* DB src */
      mendm),          /* ENDM */

      /* MOV dst:Ew,src:Db */
mov2    cmac14struc data(
      .mov1,2,
      specE,modw,specD,modb,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0c7h,        /* DB 0C7h */
      mmodrm1,0,dst,    /* MODRM 0,dst */
      mdwf,src,         /* DW src */
      mendm),          /* ENDM */

      /* MOV dst:Ew,src:Dw */
mov3    cmac14struc data(
      .mov2,2,
      specE,modw,specD,modw,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0c7h,        /* DB 0C7h */
      mmodrm1,0,dst,    /* MODRM 0,dst */
      mdwf,src,         /* DW src */
      mendm),          /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

mov4      /* MOV dst:Rb,src:Db */
          cmac16struc data(
            .mov3,2,
            specR,modb,specD,modb,
            mdbit,          /* DBIT 5(101108),3(dst(0)) */
            mnumberbits,5,16h,
            mformalbits,3,dst,0,
            mendbit,
            mdbf,src,        /* DB src */
            mendum),        /* ENDM */

```

```

mov5      /* MOV dst:Rw,src:Db */
          cmac16struc data(
            .mov4,2,
            specR,modw,specD,modb,
            mdbit,          /* DBIT 5(101118),3(dst(0)) */
            mnumberbits,5,17h,
            mformalbits,3,dst,0,
            mendbit,
            mdwf,src,        /* DW src */
            mendum),        /* ENDM */

```

```

mov5      /* MOV dst:Rw,src:Dw */
          cmac16struc data(
            .mov5,2,
            specR,modw,specD,modw,
            mdbit,          /* DBIT 5(101118),3(dst(0)) */
            mnumberbits,5,17h,
            mformalbits,3,dst,0,
            mendbit,
            mdwf,src,        /* DW src */
            mendum),        /* ENDM */

```

```

mov7      /* MOV dst:Eb,src:Rb */
          cmac12struc data(
            .mov6,2,
            specE,modb,specR,modb,
            msegfix,dst,     /* SEGFIX dst */
            mdbn,88h,        /* DB 88H */
            mmodrm2,src,dst, /* MODRM src,dst */
            mendum),        /* ENDM */

```

```

mov8      /* MOV dst:Ew,src:Rw */
          cmac12struc data(
            .mov7,2,
            specE,modw,specR,modw,
            msegfix,dst,     /* SEGFIX dst */
            mdbn,89h,        /* DB 89H */
            mmodrm2,src,dst, /* MODRM src,dst */
            mendum),        /* ENDM */

```

```

mov9      /* MOV dst:Rb,src:Eb */
          cmac12struc data(
            .mov8,2,
            specR,modb,specE,modb,
            msegfix,src,     /* SEGFIX src */
            mdbn,8ah,        /* DB 8AH */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mmodrm2,dst,src,      /* MODRM dst,src */
        mendum),             /* ENDM */

/* MOV dst:Rw,src:Ew */
mov10   cmac12struc data(
        .mov9,2,
        specR,modw,specE,modw,
        msegfix,src,         /* SEGFIX src */
        mdbn,8bh,           /* DB 8Bh */
        mmodrm2,dst,src,     /* MODRM dst,src */
        mendum),            /* ENDM */

/* MOV dst:Ew,src:S */
mov11   cmac12struc data(
        .mov10,2,
        specE,modw,specS,nomod,
        msegfix,dst,         /* SEGFIX dst */
        mdbn,8ch,           /* DB 8Ch */
        mmodrm2,src,dst,     /* MODRM src,dst */
        mendum),            /* ENDM */

/* MOV dst:Sd(ES),src:Ew */
mov12   cmac13struc data(
        .mov11,2,
        specS,modd+singlerange+registerrange,res,
        specE,modw,
        msegfix,src,         /* SEGFIX src */
        mdbn,8eh,           /* DB 8Eh */
        mmodrm2,dst,src,     /* MODRM dst,src */
        mendum),            /* ENDM */

/* MOV dst:Sd(SS,DS),src:Ew */
mov13   cmac14struc data(
        .mov12,2,
        specS,modd+doublerange+registerrange,rss,rds,
        specE,modw,
        msegfix,src,         /* SEGFIX src */
        mdbn,8eh,           /* DB 8Eh */
        mmodrm2,dst,src,     /* MODRM dst,src */
        mendum),            /* ENDM */

/* MOV dst:Ab,src:Xb */
mov14   cmac11struc data(
        .mov13,2,
        specA,modb,specX,modb,
        msegfix,src,         /* SEGFIX src */
        mdbn,0a0h,          /* DB 0A0h */
        mdwf,src,            /* DW src */
        mendum),            /* ENDM */

/* MOV dst:Aw,src:Xw */
mov15   cmac11struc data(
        .mov14,2,
        specA,modw,specX,modw,
        msegfix,src,         /* SEGFIX src */
        mdlb,0a1h,          /* DB 0A1h */
        mdwf,src,            /* DW src */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mendum),          /* END4 */

/* MOV dst:Xb,src:Ab */
movl6   cmac11struc data(
        .movl5,2,
        specX,modb,specA,modb,
        msegfix,dst,      /* SEGFIX dst */
        mdbn,0a2h,        /* DB 0A2H */
        mdwf,dst,         /* DW dst */
        mendum),          /* ENDM */

/* MOV dst:Xw,src:Aw */
movl7   cmac11struc public data(
        .movl6,2,
        specX,modw,specA,modw,
        msegfix,dst,      /* SEGFIX dst */
        mdbn,0a3h,        /* DB 0A3H */
        mdwf,dst,         /* DW dst */
        mendum),          /* ENDM */

/* MOVS SI$ptr:Eb,DI$ptr:Eb */
movs1   cmac12struc data(
        nil,2,
        specE,modb,specE,modb,
        mnosegfix,res,si$ptr, /* NOSEGFIX ES,SI$ptr */
        msegfix,di$ptr,      /* SEGFIX DI$ptr */
        mdbn,0a4h,          /* DB 0A4H */
        mendum),            /* ENDM */

/* MOVS SI$ptr:Ew,DI$ptr:Ew */
movs2   cmac12struc public data(
        .movs1,2,
        specE,modw,specE,modw,
        mnosegfix,res,si$ptr, /* NOSEGFIX ES,SI$ptr */
        msegfix,di$ptr,      /* SEGFIX DI$ptr */
        mdbn,0a5h,          /* DB 0A5H */
        mendum),            /* ENDM */

/* MOVSB */
MOVSB1   CMAC3STRUC PUBLIC DATA(
        NIL,0,
        MODBN,0A4H,
        MENDM),

/* MOVSW */
MOVSW1   CMAC3STRUC PUBLIC DATA(
        NIL,0,
        MODBN,0A5H,
        MENDM),

/* MUL mplier:Eb */
mull     cmac10struc data(
        nil,1,
        specE,modb,
        msegfix,mplier,      /* SEGFIX mplier */
        mdbn,0f6h,          /* DB 0F6H */
        mmodrml,4,mplier,    /* MODRM 4,mplier */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

mendm),          /* ENDM */

/* MUL mplier:Lw */
mul2 cmac10struc public data(
    .mul1,1,
    specE,modw,
    msegfix,mplier,    /* SEGFIX mplier */
    mdbn,0f7h,        /* DB 0F7H */
    mmodrm1,4,mplier,  /* MODRM 4,mplier */
    mendm),          /* ENDM */

/* NEG dst:Eb */
neg1 cmac10struc data(
    nil,1,
    specE,modb,
    msegfix,dst,        /* SEGFIX dst */
    mdbn,0f6h,        /* DB 0F6H */
    mmodrm1,3,dst,      /* MODRM 3,dst */
    mendm),          /* ENDM */

/* NEG dst:Ew */
neg2 cmac10struc public data(
    .neg1,1,
    specE,modw,
    msegfix,dst,        /* SEGFIX dst */
    mdbn,0f7h,        /* DB 0F7H */
    mmodrm1,3,dst,      /* MODRM 3,dst */
    mendm),          /* ENDM */

/* NOP */
nop1 cmac3struc public data(
    nil,0,
    mdbn,90h,          /* DB 90H */
    mendm),          /* ENDM */

/* NOT dst:Eb */
not1 cmac10struc data(
    nil,1,
    specE,modb,
    msegfix,dst,        /* SEGFIX dst */
    mdbn,0f6h,        /* DB 0F6H */
    mmodrm1,2,dst,      /* MODRM 2,dst */
    mendm),          /* ENDM */

/* NOT dst:Ew */
not2 cmac10struc public data(
    .not1,1,
    specE,modw,
    msegfix,dst,        /* SEGFIX dst */
    mdbn,0f7h,        /* DB 0F7H */
    mmodrm1,2,dst,      /* MODRM 2,dst */
    mendm),          /* ENDM */

/* OR dst:Eb,src:Db */
or1 cmac14struc data(
    nil,2,
    specE,modb,specD,modb,

```

CCP/M-86 2-0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93350
 SER. # CC-104

```

msegfix,dst,      /* SEGFIX dst */
mdbn,80h,         /* DB 80h */
mmodrm1,1,dst,    /* MODRM 1,dst */
mdbf,src,         /* DB src */
mendm),          /* ENDM */

/* OR dst:Ew,src:Dw */
or2 cmac14struc data(
    .or1,2,
    specE,modw,specD,modw,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,81h,         /* DB 81h */
    mmodrm1,1,dst,    /* MODRM 1,dst */
    mdwf,src,         /* DW src */
    mendm),          /* ENDM */

/* OR dst:Ew,src:Db */
or3 cmac14struc data(
    .or2,2,
    specE,modw,specD,modb,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,81h,         /* DB 81h */
    mmodrm1,1,dst,    /* MODRM 1,dst */
    mdwf,src,         /* DW src */
    mendm),          /* ENDM */

/* OR dst:Ab,src:Db */
or4 cmac9struc data(
    .or3,2,
    specA,modb,specD,modb,
    mdbn,0ch,         /* DB 0CH */
    mdbf,src,         /* DB src */
    mendm),          /* ENDM */

/* OR dst:Aw,src:Db */
or5 cmac9struc data(
    .or4,2,
    specA,modw,specD,modb,
    mdbn,0dh,         /* DB 0DH */
    mdwf,src,         /* DW src */
    mendm),          /* ENDM */

/* OR dst:Aw,src:Dw */
or6 cmac9struc data(
    .or5,2,
    specA,modw,specD,modw,
    mdbn,0dh,         /* DB 0DH */
    mdwf,src,         /* DW src */
    mendm),          /* ENDM */

/* OR dst:Eb,src:Rb */
or7 cmac12struc data(
    .or6,2,
    specE,modb,specR,modb,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,08h,         /* DB 08h */
    mmodrm2,src,dst,  /* MODRM src,dst */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

mendm),          /* ENDM */

/* OR dst:Rw,src:Rw */
or8 cmac12struc data(
    .or7,2,
    specE,modw,specR,modw,
    msegfix,dst,          /* SEGfix dst */
    mdbn,09h,             /* DB 09h */
    mmodrm2,src,dst,      /* MODRM src,dst */
    mendm),              /* ENDM */

/* OR dst:Rb,src:Eb */
or9 cmac12struc data(
    .or8,2,
    specR,modb,specE,modb,
    msegfix,src,          /* SEGfix src */
    mdbn,0ah,             /* DB 0Ah */
    mmodrm2,dst,src,      /* MODRM dst,src */
    mendm),              /* ENDM */

/* OR dst:Rw,src:Ew */
or10 cmac12struc public data(
    .or9,2,
    specR,modw,specE,modw,
    msegfix,src,          /* SEGfix src */
    mdbn,0bh,             /* DB 0Bh */
    mmodrm2,dst,src,      /* MODRM dst,src */
    mendm),              /* ENDM */

/* OUT dst:Db,src:Ab */
out1 cmac9struc data(
    nil,2,
    specD,modb,specA,modb,
    mdbn,0e6h,            /* DB 0E6H */
    mdbf,dst,             /* DB dst */
    mendm),              /* ENDM */

/* OUT dst:Db,src:Aw */
out2 cmac9struc data(
    .out1,2,
    specD,modb,specA,modw,
    mdbn,0e7h,            /* DB 0E7H */
    mdbf,dst,             /* DB dst */
    mendm),              /* ENDM */

/* OUT dst:Rw(DX),src:Ab */
out3 cmac8struc data(
    .out2,2,
    specR,modw+singlerange+registerrange,rdx,
    specA,modb,
    mdbn,0eeh,            /* DB 0EEH */
    mendm),              /* ENDM */

/* OUT dst:Rw(DX),src:Aw */
out4 cmac8struc public data(
    .out3,2,
    specR,modw+singlerange+registerrange,rdx,

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

specA,modw,
mdbn,0efh,      /* DB 0EFH */
mendm),         /* ENDM */

/* POP dst:Ew */
pop1 cmac10struc data(
nil,1,
specE,modw,
msejfix,dst,    /* SEGFIX dst */
mdbn,8fh,       /* DB 8FH */
mmodrm1,0,dst,  /* MODRM 0,dst */
mendm),        /* ENDM */

/* POP dst:Sd(LS) */
pop2 cmac16struc data(
.pop1,1,
specS,modd+singlerange+registerrange,res,
mdbit,          /* DBIT 3(0),2(dst(0)),3(7) */
mnumberbits,3,0,
mformalbits,2,dst,0,
mnumberbits,3,7,
mendbit,
mendm),        /* ENDM */

/* POP dst:Sd(SS,DS) */
pop3 cmac17struc data(
.pop2,1,
specS,modd+doublerange+registerrange,rss,rds,
mdbit,          /* DBIT 3(0),2(dst(0)),3(7) */
mnumberbits,3,0,
mformalbits,2,dst,0,
mnumberbits,3,7,
mendbit,
mendm),        /* ENDM */

/* POP dst:Rw */
pop4 cmac12struc public data(
.pop3,1,
specR,modw,
mdbit,          /* DBIT 5(01011b),3(dst(0)) */
mnumberbits,5,0bh,
mformalbits,3,dst,0,
mendbit,
mendm),        /* ENDM */

/* POPF */
popf1 cmac3struc public data(
nil,0,
mdbn,9dh,       /* DB 9DH */
mendm),         /* ENDM */

/* PUSH dst:Ew */
push1 cmac10struc data(
nil,1,
specE,modw,
msejfix,dst,    /* SEGFIX dst */
mdbn,0ffh,      /* DB 0FFH */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

        mmodrml,6,dst,      /* MODRM 6,dst */
        mendum),          /* ENDM */

/* PUSH dst:Sd */
push2   cmac15struc data(
        .pushl,1,
        specS,modd,
        mdbit,             /* DBIT 3(0),2(dst(0)),3(6) */
        mnumberbits,3,0,
        mformalbits,2,dst,0,
        mnumberbits,3,6,
        mndbit,
        mendum),          /* ENDM */

/* PUSH dst:Rw */
push3   cmac12struc public data(
        .push2,1,
        specR,modw,
        mdbit,             /* DBIT 5(01010B),3(dst(0)) */
        mnumberbits,5,0ah,
        mformalbits,3,dst,0,
        mndbit,
        mendum),          /* ENDM */

/* PUSHf */
pushf1  cmac3struc public data(
        nil,0,
        mdbn,9ch,          /* DB 9CH */
        mendum);          /* ENDM */

```

```

19 1      end$module cmac3;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

MODULE INFORMATION:

```

CODE AREA SIZE      = 0000H      0D
CONSTANT AREA SIZE  = 02CCH      716D
VARIABLE AREA SIZE  = 0000H      0D
MAXIMUM STACK SIZE  = 0000H      0D
780 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE CMAC4

OBJECT MODULE PLACED IN CMAC4.OBJ

COMPILER INVOKED BY: :F0: CMAC4.PLM DEBUG OPTIMIZE(2) DATE(10/5/81) PAGES(100)

```

1      $title ('CODEMACRO DEFINITIONS - PART 4')
      cmac4:
      do;

      /*

      modified 6/16/81 R. Silberstein

      */

      $include (:f1:macro.lit)
=      $nolist
      $include (:f1:cmacd.lit)
=      $nolist
      $include (:f1:equals.lit)
=      $nolist
      $include (:f1:cmac.lit)
=      $nolist

18 1    dcl

      /* RCL dst:Eb,count:Db(1) */
rcl1   cmac13struc data(
      nil,2,
      specE,modb,
      specD,modb+singlerange+number$range,1,
      msegfix,dst,          /* SEGFIX dst */
      mdbn,0d0h,           /* DB 0D0H */
      mmodrm1,2,dst,       /* MODRM 2,dst */
      mendm),              /* ENDM */

      /* RCL dst:Ew,count:Db(1) */
rcl2   cmac13struc data(
      .rcl1,2,
      specE,modw,
      specD,modb+singlerange+number$range,1,
      msegfix,dst,          /* SEGFIX dst */
      mdbn,0d1h,           /* DB 0D1H */
      mmodrm1,2,dst,       /* MODRM 2,dst */
      mendm),              /* ENDM */

      /* RCL dst:Eb,count:Rb(CL) */
rcl3   cmac13struc data(
      .rcl2,2,
      specE,modb,
      specR,modb+singlerange+register$range,rcl,
      msegfix,dst,          /* SEGFIX dst */
      mdbn,0d2h,           /* DB 0D2H */
      mmodrm1,2,dst,       /* MODRM 2,dst */
      mendm),              /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

/* RCL dst:Ew,count:Rb(CCL) */
rc14  cmac13struc public data(
      .rc13,2,
      specE,modw,
      specR,modb+singlerange+register$range,rc1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d3h,        /* DB 0D3H */
      mmodrm1,2,dst,    /* MODRM 2,dst */
      mendm),           /* ENDM */

/* RCR dst:Eb,count:Db(1) */
rcr1  cmac13struc data(
      nil,2,
      specE,modb,
      specD,modb+singlerange+number$range,1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d0h,        /* DB 0D0H */
      mmodrm1,3,dst,    /* MODRM 3,dst */
      mendm),           /* ENDM */

/* RCR dst:Ew,count:Db(1) */
rcr2  cmac13struc data(
      .rcr1,2,
      specE,modw,
      specD,modb+singlerange+number$range,1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d1h,        /* DB 0D1H */
      mmodrm1,3,dst,    /* MODRM 3,dst */
      mendm),           /* ENDM */

/* RCR dst:Eb,count:Rb(CCL) */
rcr3  cmac13struc data(
      .rcr2,2,
      specE,modb,
      specR,modb+singlerange+register$range,rc1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d2h,        /* DB 0D2H */
      mmodrm1,3,dst,    /* MODRM 3,dst */
      mendm),           /* ENDM */

/* RCR dst:Ew,count:Rb(CCL) */
rcr4  cmac13struc public data(
      .rcr3,2,
      specE,modw,
      specR,modb+singlerange+register$range,rc1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d3h,        /* DB 0D3H */
      mmodrm1,3,dst,    /* MODRM 3,dst */
      mendm),           /* ENDM */

/* REP PREFIX */
repl  cmac3struc public data(
      nil,0+prefix$on,
      mdbn,0f3h,        /* DB 0F3H */
      mendm),           /* ENDM */

/* REPE PREFIX */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

repel    cmac3struc public data(
          nil,0+prefix$on,
          mdbn,0f3h,          /* DB 0F3H */
          mendum),          /* ENDM */

          /* REPNE PREFIX */
repnel    cmac3struc public data(
          nil,0+prefix$on,
          mdbn,0f2h,          /* DB 0F2H */
          mendum),          /* ENDM */

          /* RET dst:Db */
retl      cmac7struc data(
          nil,1,
          spec0,modb,
          mdbn,0c2h,          /* DB 0C2H */
          mdwf,dst,          /* DW dst */
          mendum),          /* ENDM */

          /* RET dst:Dw */
ret2      cmac7struc data(
          .retl,1,
          spec0,modw,
          mdbn,0c2h,          /* DB 0C2H */
          mdwf,dst,          /* DW dst */
          mendum),          /* ENDM */

          /* RET */
ret3      cmac3struc public data(
          .ret2,0,
          mdbn,0c3h,          /* DB 0C3H */
          mendum),          /* ENDM */

          /* RETF dst:Db */
retf1     cmac7struc data(
          nil,1,
          spec0,modb,
          mdbn,0cah,          /* DB 0CAH */
          mdwf,dst,          /* DW dst */
          mendum),          /* ENDM */

          /* RETF dst:Dw */
retf2     cmac7struc data(
          .retf1,1,
          spec0,modw,
          mdbn,0cah,          /* DB 0CAH */
          mdwf,dst,          /* DW dst */
          mendum),          /* ENDM */

          /* RETF */
retf3     cmac3struc public data(
          .retf2,0,
          mdbn,0cbh,          /* DB 0CBH */
          mendum),          /* ENDM */

          /* ROL dst:Eb,count:)b(1) */
roll      cmac13struc data(

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

nil,2,
specE,mod0,
spec0,modb+singlerange+number$range,1,
msegfix,dst,      /* SEGFIX dst */
mdbn,0d0h,        /* DB 0D0H */
mmodrm1,0,dst,    /* MODRM 0,dst */
mendm),          /* ENDM */

/* ROL dst:Eb,count:3b(1) */
rol2 cmac13struc data(
    .rol1,2,
    specE,modw,
    spec0,modb+singlerange+number$range,1,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,0d1h,        /* DB 0D1H */
    mmodrm1,0,dst,    /* MODRM 0,dst */
    mendm),          /* ENDM */

/* ROL dst:Eb,count:3b(CL) */
rol3 cmac13struc data(
    .rol2,2,
    specE,modb,
    specR,modb+singlerange+register$range,rcl,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,0d2h,        /* DB 0D2H */
    mmodrm1,0,dst,    /* MODRM 0,dst */
    mendm),          /* ENDM */

/* ROL dst:Eb,count:3b(CL) */
rol4 cmac13struc public data(
    .rol3,2,
    specE,modw,
    specR,modb+singlerange+register$range,rcl,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,0d3h,        /* DB 0D3H */
    mmodrm1,0,dst,    /* MODRM 0,dst */
    mendm),          /* ENDM */

/* ROR dst:Eb,count:3b(1) */
ror1 cmac13struc data(
    nil,2,
    specE,modb,
    spec0,modb+singlerange+number$range,1,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,0d0h,        /* DB 0D0H */
    mmodrm1,1,dst,    /* MODRM 1,dst */
    mendm),          /* ENDM */

/* ROR dst:Eb,count:3b(1) */
ror2 cmac13struc data(
    .ror1,2,
    specE,modw,
    spec0,modb+singlerange+number$range,1,
    msegfix,dst,      /* SEGFIX dst */
    mdbn,0d1h,        /* DB 0D1H */
    mmodrm1,1,dst,    /* MODRM 1,dst */
    mendm),          /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

/* ROR dst:Eb,count:Rb(CL) */
ror3  cmac13struc data(
      .ror2,2,
      specE,modb,
      specR,modb+singlerange+register$range,rcl,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d2h,        /* DB 0D2H */
      mmodrm1,1,dst,    /* MODRM 1,dst */
      mendm),           /* ENDM */

/* ROR dst:Ew,count:Rb(CL) */
ror4  cmac13struc public data(
      .ror3,2,
      specE,modw,
      specR,modb+singlerange+register$range,rcl,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d3h,        /* DB 0D3H */
      mmodrm1,1,dst,    /* MODRM 1,dst */
      mendm),           /* ENDM */

/* SAHF */
sahf1  cmac3struc public data(
      nil,0,
      mdbn,9eh,         /* DB 9eh */
      mendm),           /* ENDM */

/* SAL dst:Eb,count:Db(1) */
sall   cmac13struc data(
      nil,2,
      specE,modb,
      specD,modb+singlerange+number$range,1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d0h,        /* DB 0D0H */
      mmodrm1,4,dst,    /* MODRM 4,dst */
      mendm),           /* ENDM */

/* SAL dst:Ew,count:Db(1) */
sal2   cmac13struc data(
      .sall,2,
      specE,modw,
      specD,modb+singlerange+number$range,1,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d1h,        /* DB 0D1H */
      mmodrm1,4,dst,    /* MODRM 4,dst */
      mendm),           /* ENDM */

/* SAL dst:Eb,count:Rb(CL) */
sal3   cmac13struc data(
      .sal2,2,
      specE,modb,
      specR,modb+singlerange+register$range,rcl,
      msegfix,dst,      /* SEGFIX dst */
      mdbn,0d2h,        /* DB 0D2H */
      mmodrm1,4,dst,    /* MODRM 4,dst */
      mendm),           /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

sal4      /* SAL dst:Ew,count:Rb(CL) */
          cmac13struc public data(
            .sal3,2,
            specE,modw,
            specR,modb+singlerange+register$range,rcl,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,0d3h,        /* DB 003H */
            mmodrm1,4,dst,    /* MODRM 4,dst */
            mendum),          /* ENDM */

```

```

sar1      /* SAR dst:Eb,count:Db(1) */
          cmac13struc data(
            nil,2,
            specE,modb,
            specD,modb+singlerange+number$range,1,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,0d0h,        /* DB 000H */
            mmodrm1,7,dst,    /* MODRM 7,dst */
            mendum),          /* ENDM */

```

```

sar2      /* SAR dst:Ew,count:Db(1) */
          cmac13struc data(
            .sar1,2,
            specE,modw,
            specD,modb+singlerange+number$range,1,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,0d1h,        /* DB 001H */
            mmodrm1,7,dst,    /* MODRM 7,dst */
            mendum),          /* ENDM */

```

```

sar3      /* SAR dst:Eb,count:Db(CL) */
          cmac13struc data(
            .sar2,2,
            specE,modb,
            specR,modb+singlerange+register$range,rcl,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,0d2h,        /* DB 002H */
            mmodrm1,7,dst,    /* MODRM 7,dst */
            mendum),          /* ENDM */

```

```

sar4      /* SAR dst:Ew,count:Rb(CL) */
          cmac13struc public data(
            .sar3,2,
            specE,modw,
            specR,modb+singlerange+register$range,rcl,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,0d3h,        /* DB 003H */
            mmodrm1,7,dst,    /* MODRM 7,dst */
            mendum),          /* ENDM */

```

```

sbb1      /* SBB dst:Eb,src:Db */
          cmac14struc data(
            nil,2,
            specE,modb,
            specD,modb,
            msegfix,dst,      /* SEGFIX dst */
            mdbn,80h,         /* DB 80H */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

mmodrm1,3,dst,      /* MODRM 3,dst */
mddf,src,           /* DB src */
mendm),             /* ENDM */

/* SBB dst:Ew,src:Dsb */
sbb2 cmac14struc data(
.sbb1,2,
specE,modw,
specD,modb,
msegfix,dst,        /* SEGFIX dst */
mdbn,8lh,           /* DB 8lh */
mmodrm1,3,dst,      /* MODRM 3,dst */
mdwf,src,           /* DW src */
mendm),             /* ENDM */

/* SBB dst:Ew,src:Dsb */
sbb3 cmac14struc data(
.sbb2,2,
specE,modw,
specD,modsb,
msegfix,dst,        /* SEGFIX dst */
mdbn,83h,           /* DB 83H */
mmodrm1,3,dst,      /* MODRM 3,dst */
mddf,src,           /* DB src */
mendm),             /* ENDM */

/* SBB dst:Ew,src:Dw */
sbb4 cmac14struc data(
.sbb3,2,
specE,modw,
specD,modw,
msegfix,dst,        /* SEGFIX dst */
mdbn,8lh,           /* DB 8lh */
mmodrm1,3,dst,      /* MODRM 3,dst */
mdwf,src,           /* DW src */
mendm),             /* ENDM */

/* SBB dst:Ab,src:Dsb */
sbb5 cmac9struc data(
.sbb4,2,
specA,modb,
specD,modb,
mdbn,1ch,           /* DB 1CH */
mddf,src,           /* DB src */
mendm),             /* ENDM */

/* SBB dst:Aw,src:Dsb */
sbb6 cmac9struc data(
.sbb5,2,
specA,modw,
specD,modb,
mdbn,ldh,           /* DB 10H */
mdwf,src,           /* DW src */
mendm),             /* ENDM */

/* SBB dst:Aw,src:Dw */
sbb7 cmac9struc data(

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

.sbb6,2,
specA,modw,
specD,modw,
mdbn,1ah,      /* DB 10H */
mdwf,src,      /* DW src */
mendm),        /* ENDM */

/* SBB dst:Eb,src:Rb */
sbb8 cmac12struc data(
.sbb7,2,
specE,modb,
specR,modb,
msegfix,dst,   /* SEGFIX dst */
mdbn,18h,      /* DB 18H */
mmodrm2,src,dst, /* MODRM src,dst */
mendm),        /* ENDM */

/* SBB dst:Ew,src:Rw */
sbb9 cmac12struc data(
.sbb8,2,
specE,modw,
specR,modw,
msegfix,dst,   /* SEGFIX dst */
mdbn,19h,      /* DB 19H */
mmodrm2,src,dst, /* MODRM src,dst */
mendm),        /* ENDM */

/* SBB dst:Rb,src:Eb */
sbb10 cmac12struc data(
.sbb9,2,
specR,modb,
specE,modb,
msegfix,src,   /* SEGFIX src */
mdbn,1ah,      /* DB 1AH */
mmodrm2,dst,src, /* MODRM dst,src */
mendm),        /* ENDM */

/* SBB dst:Rw,src:Ew */
sbb11 cmac12struc public data(
.sbb10,2,
specR,modw,
specE,modw,
msegfix,src,   /* SEGFIX src */
mdbn,1bh,      /* DB 1BH */
mmodrm2,dst,src, /* MODRM dst,src */
mendm),        /* ENDM */

/* SCAS dst:Eb */
scas1 cmac8struc data(
nil,1,
specE,modb,
mnosegfix,res,dst, /* NOSEGFIX ES,dst */
mdbn,0aeh,      /* DB 0AEH */
mendm),        /* ENDM */

/* SCAS dst:Ew */
scas2 cmac8struc public data(

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

        .scas1,1,
        specE,modw,
        mnosegfix,reg,dst, /* NOSEGFIX ES,dst */
        mdbn,0afh, /* DB 0AFH */
        mendm), /* ENDM */

/* SCASB */
SCASB1 CMAC3STRUC PUBLIC DATA
NIL,0,
MDBN,0AFH,
MENDM),

/* SCASW */
SCASW1 CMAC3STRUC PUBLIC DATA
NIL,0,
MDBN,0AFH,
MENDM),

/* SHR dst:Eb,count:Jb(1) */
shr1 cmac13struc data(
nil,2,
specE,modb,
specD,modb+singlerange+number$range,1,
msegfix,dst, /* SEGFIX dst */
mdbn,0d0h, /* DB 0D0H */
mmodrm1,5,dst, /* MODRM 5,dst */
mendm), /* ENDM */

/* SHR dst:Fw,count:Jb(1) */
shr2 cmac13struc data(
.shr1,2,
specE,modw,
specD,modb+singlerange+number$range,1,
msegfix,dst, /* SEGFIX dst */
mdbn,0d1h, /* DB 0D1H */
mmodrm1,5,dst, /* MODRM 5,dst */
mendm), /* ENDM */

/* SHR dst:Eb,count:Rb(CL) */
shr3 cmac13struc data(
.shr2,2,
specE,modb,
specR,modb+singlerange+register$range,rcl,
msegfix,dst, /* SEGFIX dst */
mdbn,0d2h, /* DB 0D2H */
mmodrm1,5,dst, /* MODRM 5,dst */
mendm), /* ENDM */

/* SHR dst:Fw,count:Rb(CL) */
shr4 cmac13struc public data(
.shr3,2,
specE,modw,
specR,modb+singlerange+register$range,rcl,
msegfix,dst, /* SEGFIX dst */
mdbn,0d3h, /* DB 0D3H */
mmodrm1,5,dst, /* MODRM 5,dst */
mendm); /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

19 1 end\$module cmac4;

MODULE INFORMATION:

CODE AREA SIZE	= 0000H	00
CONSTANT AREA SIZE	= 02D2H	722D
VARIABLE AREA SIZE	= 0000H	00
MAXIMUM STACK SIZE	= 0000H	00
808 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93350

SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE CMACS
 OBJECT MODULE PLACED IN CMACS.OBJ
 COMPILER INVOKED BY: :F0: CMACS.PLM DEBJG OPTIMIZE(2) DATE(10/5/81) PAGESWIDTH(100)

```
$title ("CODEMACRO DEFINITIONS - PART 5")
```

```
cmac5:
```

```
do;
```

```
/*
```

```
modified 6/16/81 R. Silberstein
```

```
*/
```

```
$include (:f1:macro.lit)
```

```
= $nolist
```

```
$include (:f1:cmacd.lit)
```

```
= $nolist
```

```
$include (:f1:equals.lit)
```

```
= $nolist
```

```
$include (:f1:cmac.lit)
```

```
= $nolist
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
18 1 dcl
```

```
/* SIC */
```

```
stcl cmac3struc public data(
```

```
nil,0,
```

```
mdbn,0f9h,
```

```
/* DB 0F9H */
```

```
mendm),
```

```
/* ENDM */
```

```
/* STD */
```

```
stdl cmac3struc public data(
```

```
nil,0,
```

```
mdbn,0fdh,
```

```
/* DB 0FDH */
```

```
mendm),
```

```
/* ENDM */
```

```
/* SII */
```

```
stil cmac3struc public data(
```

```
nil,0,
```

```
mdbn,0fbh,
```

```
/* DB 0FBH */
```

```
mendm),
```

```
/* ENDM */
```

```
/* STOS dst:eb */
```

```
stosl cmac8struc data(
```

```
nil,1,
```

```
specE,modb,
```

```
mnosegfix,res,dst, /* NOSEGFIX ES,dst */
```

```
mdbn,0aah,
```

```
/* DB 0AAH */
```

```
mendm),
```

```
/* ENDM */
```

```
/* STOS dst:lw */
```

```
stos2 cmac8struc public data(
```

```
.stosl,1,
```

```
specE,modw,
```

```

mnosegfix,res,dst, /* NOSEGFIX ES,dst */
mdbh,0dbn, /* DB 04BH */
mendm), /* ENDM */

/* STOSB */
CMAC3STRUC PUBLIC DATA1
NIL,0,
MDBN,0AAH,
MENDM),

/* STOSW */
CMAC3STRUC PUBLIC DATA2
NIL,0,
MDBN,0ABH,
MENDM),

/* SUB dst:Eb,src:Db */
sub1 cmac14struc data(
nil,2,
specE,modb,
specD,modb,
msegfix,dst, /* SEGFIX dst */
mdbn,80h, /* DB 80H */
mmodrm1,5,dst, /* MODRM 5,dst */
mdbf,src, /* DB src */
mendm), /* ENDM */

/* SUB dst:Ew,src:Db */
sub2 cmac14struc data(
.sub1,2,
specE,modw,
specD,modb,
msegfix,dst, /* SEGFIX dst */
mdbn,81h, /* DB 81H */
mmodrm1,5,dst, /* MODRM 5,dst */
mdwf,src, /* DW src */
mendm), /* ENDM */

/* SUB dst:Ew,src:Db */
sub3 cmac14struc data(
.sub2,2,
specE,modw,
specD,modsb,
msegfix,dst, /* SEGFIX dst */
mdbn,83h, /* DB 83H */
mmodrm1,5,dst, /* MODRM 5,dst */
mdbf,src, /* DB src */
mendm), /* ENDM */

/* SUB dst:Ew,src:Dw */
sub4 cmac14struc data(
.sub3,2,
specE,modw,
specD,modw,
msegfix,dst, /* SEGFIX dst */
mdbn,81h, /* DB 81H */
mmodrm1,5,dst, /* MODRM 5,dst */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        mdwf,src,          /* DW src */
        mendum),          /* ENDA */

/* SUB dst:Ab,src:Db */
sub5   cmac9struc data(
        .sub4,2,
        specA,modb,
        specD,modb,
        mdbn,2ch,          /* DB 2CH */
        mdbf,src,          /* DB src */
        mendum),          /* ENDA */

/* SUB dst:Aw,src:Db */
sub6   cmac9struc data(
        .sub5,2,
        specA,modw,
        specD,modb,
        mdbn,2dh,          /* DB 2DH */
        mdwf,src,          /* DW src */
        mendum),          /* ENDA */

/* SUB dst:Aw,src:Dw */
sub7   cmac9struc data(
        .sub6,2,
        specA,modw,
        specD,modw,
        mdbn,2dh,          /* DB 2DH */
        mdwf,src,          /* DW src */
        mendum),          /* ENDA */

/* SUB dst:Eb,src:Rb */
sub8   cmac12struc data(
        .sub7,2,
        specE,modb,
        specR,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,28h,          /* DB 28H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDA */

/* SUB dst:Ew,src:Rw */
sub9   cmac12struc data(
        .sub8,2,
        specE,modw,
        specR,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,29h,          /* DB 29H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDA */

/* SUB dst:Rb,src:Eb */
sub10  cmac12struc data(
        .sub9,2,
        specR,modb,
        specE,modb,
        msegfix,src,       /* SEGFIX src */
        mdbn,2ah,          /* DB 2AH */

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

        mmodrm2,dst,src,    /* MODRM dst,src */
        mendum),          /* ENDM */

/* SUB dst:Rw,src:Ew */
sub11  cmac12struc public data(
        .sub10,2,
        specE,modw,
        specE,modw,
        msegfix,src,       /* SEGFIX src */
        mdbn,2bh,         /* DB 2BH */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

/* TEST dst:Eb,src:Db */
test1  cmac14struc data(
        nil,2,
        specE,modb,
        specD,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,0f6h,         /* DB 0F6H */
        mmodrm1,0,dst,     /* MODRM 0,dst */
        mdbf,src,          /* DB src */
        mendum),          /* ENDM */

/* TEST dst:Ew,src:Db */
test2  cmac14struc data(
        .test1,2,
        specE,modw,
        specD,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,0f7h,         /* DB 0F7H */
        mmodrm1,0,dst,     /* MODRM 0,dst */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* TEST dst:Ew,src:Dw */
test3  cmac14struc data(
        .test2,2,
        specE,modw,
        specD,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,0f7h,         /* DB 0F7H */
        mmodrm1,0,dst,     /* MODRM 0,dst */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* TEST dst:Ab,src:Db */
test4  cmac9struc data(
        .test3,2,
        specA,modb,
        specD,modb,
        mdbn,0a8h,         /* DB 0A8H */
        mdbf,src,          /* DB src */
        mendum),          /* ENDM */

/* TEST dst:Aw,src:Db */
test5  cmac9struc data(

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

        .test4,2,
        specA,modw,
        specD,modb,
        mdbn,0a9h,      /* DB 0A9H */
        mdwf,src,       /* DW src */
        mendm),         /* ENDM */

/* TEST dst:Aw,src:Dw */
test6   cmac9struc data(
        .test5,2,
        specA,modw,
        specD,modw,
        mdbn,0a9h,      /* DB 0A9H */
        mdwf,src,       /* DW src */
        mendm),         /* ENDM */

/* TEST dst:Eb,src:Rb */
test7   cmac12struc data(
        .test6,2,
        specE,modb,
        specR,modb,
        msegfix,dst,     /* SEGFIX dst */
        mdbn,84h,        /* DB 84H */
        mmodrm2,src,dst, /* MODRM src,dst */
        mendm),         /* ENDM */

/* TEST dst:Ew,src:Rw */
test8   cmac12struc data(
        .test7,2,
        specE,modw,
        specR,modw,
        msegfix,dst,     /* SEGFIX dst */
        mdbn,85h,        /* DB 85H */
        mmodrm2,src,dst, /* MODRM src,dst */
        mendm),         /* ENDM */

/* TEST dst:Rb,src:Eb */
test9   cmac12struc data(
        .test8,2,
        specR,modb,
        specE,modb,
        msegfix,src,     /* SEGFIX src */
        mdbn,84h,        /* DB 84H */
        mmodrm2,dst,src, /* MODRM dst,src */
        mendm),         /* ENDM */

/* TEST dst:Rw,src:Ew */
test10  cmac12struc public data(
        .test9,2,
        specR,modw,
        specE,modw,
        msegfix,src,     /* SEGFIX src */
        mdbn,85h,        /* DB 85H */
        mmodrm2,dst,src, /* MODRM dst,src */
        mendm),         /* ENDM */

/* WATT */

```

CCP/M-86 2.0 V. 3

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104

```

wait1  cmac3struc public data(
        nil,0,
        mdbn,96h,          /* DB 96H */
        mendum),          /* ENDM */

/* XCHG dst:Eb,src:Rb */
xchg1  cmac12struc data(
        nil,2,
        specr,modb,
        specr,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,86h,         /* DB 86H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* XCHG dst:Ew,src:Rw */
xchg2  cmac12struc data(
        .xchg1,2,
        specr,modw,
        specr,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,87h,         /* DB 87H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* XCHG dst:Rb,src:Eb */
xchg3  cmac12struc data(
        .xchg2,2,
        specr,modb,
        specr,modb,
        msegfix,src,       /* SEGFIX src */
        mdbn,86h,         /* DB 86H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

/* XCHG dst:Rw,src:Ew */
xchg4  cmac12struc data(
        .xchg3,2,
        specr,modw,
        specr,modw,
        msegfix,src,       /* SEGFIX src */
        mdbn,87h,         /* DB 87H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

/* XCHG dst:Rw,src:Aw */
xchg5  cmac14struc data(
        .xchg4,2,
        specr,modw,specr,modw,
        mdbit,             /* DBIT 5(100108),3(dst(0)) */
        mnumberbits,5,12h,
        mformalbits,3,dst,0,
        mendbit,
        mendum),          /* ENDM */

/* XCHG dst:Aw,src:Rw */
xchg6  cmac14struc public data(

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        .xchg5,2,
        specA,modw,specR,modw,
        mdbit,          /* DBIT 5(100103),3(dst(0)) */
        mnumberbits,5,12h,
        mformalbits,3,src,0,
        mendbit,
        mendum),          /* ENDA */

/* XLAT dst:E */
xlat1  cmac7struc public data(
        nil,1,
        specE,nomod,
        msegfix,dst,      /* SEGFIX dst */
        mdbn,0d7h,        /* DB 0D7H */
        mendum),          /* ENDM */

/* XOR dst:Eb,src:Db */
xor1   cmac14struc data(
        nil,2,
        specE,modb,
        specD,modb,
        msegfix,dst,      /* SEGFIX dst */
        mdbn,80h,         /* DB 80H */
        mmodrm1,6,dst,    /* MODRM 6,dst */
        mdbf,src,         /* DB src */
        mendum),          /* ENDM */

/* XOR dst:Ew,src:Db */
xor2   cmac14struc data(
        .xor1,2,
        specE,modw,specD,modb,
        msegfix,dst,      /* SEGFIX dst */
        mdbn,81h,         /* DB 81H */
        mmodrm1,6,dst,    /* MODRM 6,dst */
        mdwf,src,         /* DW src */
        mendum),          /* ENDM */

/* XOR dst:Ew,src:Dw */
xor3   cmac14struc data(
        .xor2,2,
        specE,modw,specD,modw,
        msegfix,dst,      /* SEGFIX dst */
        mdbn,81h,         /* DB 81H */
        mmodrm1,6,dst,    /* MODRM 6,dst */
        mdwf,src,         /* DW src */
        mendum),          /* ENDM */

/* XOR dst:Ab,src:Db */
xor4   cmac9struc data(
        .xor3,2,
        specA,modb,specD,modb,
        mdbn,34h,         /* DB 34H */
        mdbf,src,         /* DB src */
        mendum),          /* ENDM */

/* XOR dst:Aw,src:Db */
xor5   cmac9struc data(

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

        .xor4,2,
        specA,modw,specD,modb,
        mdbn,35h,          /* DB 35H */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* XOR dst:Aw,src:Dw */
xor5    cmac9struc data(
        .xor5,2,
        specA,modw,specD,modb,
        mdbn,35h,          /* DB 35H */
        mdwf,src,          /* DW src */
        mendum),          /* ENDM */

/* XOR dst:Eb,src:Rb */
xor7    cmac12struc data(
        .xor6,2,
        specE,modb,specR,modb,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,30h,          /* DB 30H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* XOR dst:Ew,src:Rw */
xor8    cmac12struc data(
        .xor7,2,
        specE,modw,specR,modw,
        msegfix,dst,       /* SEGFIX dst */
        mdbn,31h,          /* DB 31H */
        mmodrm2,src,dst,   /* MODRM src,dst */
        mendum),          /* ENDM */

/* XOR dst:Rb,src:Eb */
xor9    cmac12struc data(
        .xor8,2,
        specR,modb,specE,modb,
        msegfix,src,       /* SEGFIX src */
        mdbn,32h,          /* DB 32H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

/* XOR dst:Rw,src:Ew */
xor10   cmac12struc public data(
        .xor9,2,
        specR,modw,specE,modw,
        msegfix,src,       /* SEGFIX src */
        mdbn,33h,          /* DB 33H */
        mmodrm2,dst,src,   /* MODRM dst,src */
        mendum),          /* ENDM */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

19 1 end\$module cmac5;

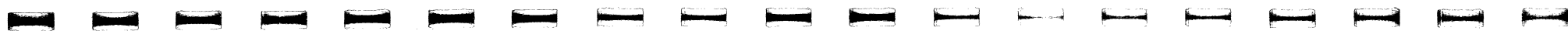
MODULE INFORMATION:

CODE AREA SIZE = 0000h 00

CONSTANT AREA SIZE = 026CH 6200
VARIABLE AREA SIZE = 0000H 00
MAXIMUM STACK SIZE = 0000H 00
743 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____



IASIS-II PL/M-86 V2.0 COMPILATION OF MODULE MNEM1

OBJECT MODULE PLACED IN MNEM1.OBJ

COMPILER INVOKED BY: :FO: MNEM1.PLH DEBJG OPTIMIZE(2) DATE(10/5/81) PAGewidth(100)

```

1      $title ("INSTRUCTION MNEMONICS MODULE - PART 1")

```

```

      mnem1:

```

```

      do;

```

```

      /*

```

```

          modified 4/10/81 R. Silberstein

```

```

          modified 6/16/81 R. Silberstein

```

```

      */

```

```

      /***** INSTRUCTION MNEMONICS *****/

```

```

      /*

```

This is all the instruction mnemonics for the assembler. The mnemonics are grouped according to the 6-bit hash value of the mnemonics - values range from 0 to 0FH.

For each instruction, there is a pointer to its code macro definition.

```

      /*

```

```

      $include (:f1:mnem.lit)

```

```

=      $nolist

```

```

      $include (:f1:cmlink.ext)

```

```

=      $nolist

```

```

      /***** MNEMONICS TABLE *****/

```

```

4      1      declare

```

```

      /*

```

```

          * HASH VALUE (HEX) *
          -----

```

```

      /*
test      opcode4      data (0,4,.test10,"TEST"),      /* 0 */
push      opcode4 public data (.test,4,.push3,"PUSH"),
SCASW     OPCOD5      DATA (0,5,.SCASW1,"SCASW"),      /* 1 */
repz      opcode4 public data (.SCASW,4,.rep1,"REPZ"),
          /* 2 */
aaa       opcode3 public data (0,3,.aaa1,"AAA"),
          /* 3 */
          /* 4 */
movs      opcode4 public data (0,4,.movs2,"MOVSB"),
daa       opcode3      data (0,3,.daa1,"DAA"),
          /* 5 */
aad       opcode3      data (.daa,3,.aad1,"AAD"),
          /* 6 */
pushf     opcode5 public data (.aad,5,.pushf1,"PUSHF"),
MOVSB     OPCOD5 PUBLIC DATA (0,5,.MOVSB1,"MOVSB"),
          /* 7 */
adc       opcode3 public data (0,3,.adc11,"ADC"),
          /* 8 */
stos      opcode4      data (0,4,.stos2,"STOS"),
          /* 9 */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

LODSW  OPCOD5      DATA (.STOS,5,.LODSW1,"LODSW"),
add     opcode3 public data (.LODSW,3,.add11,"ADD"),
CMPSW  OPCOD5 PUBLIC DATA (0,5,.CMPSW1,"CMPSW"),      /* 0a */
STOSB  OPCOD5      DATA (0,5,.STOSB1,"STOSB"),        /* 0b */
ja      opcode2 public data (.STOSB,2,.ja1,"JA"),
jb      opcode2      data (0,2,.jb1,"JB"),              /* 0c */
dec     opcode3 public data (.jb,3,.dec3,"DEC"),
JC      OPCOD2      DATA (0,2,.JB1,"JC"),              /* 0D */
loopne  opcode6 public data (.JC,6,.loopne1,"LOOPNE"),  /* 0e */
aa      opcode3      data (0,3,.aa1,"AA"),              /* 0f */
je      opcode2      data (.aa,2,.je1,"JE"),
repnz   opcode5 public data (.je,5,.repne1,"REPNZ");

```

5 1 end mnem1;

MODULE INFORMATION:

```

CODE AREA SIZE      = 0000H      0D
CONSTANT AREA SIZE  = 0002H      210D
VARIABLE AREA SIZE  = 0000H      0D
MAXIMUM STACK SIZE  = 0000H      0D
195 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE MNEM2

OBJECT MODULE PLACED IN MNEM2.OBJ

COMPILER INVOKED BY: :F0: MNEM2.PLM DEBUG OPTIMIZE(2) DATE(10/5/81) PAGERWIDTH(100)

1

```

$title ("INSTRUCTION MNEMONICS MODULE - PART 2")
mnem2:
do:

```

/*

```

    modified 4/10/81 R. Silberstein
    modified 6/16/81 R. Silberstein
    modified 7/24/81 R. Silberstein

```

*/

/***** INSTRUCTION MNEMONICS *****/

/*

This is all the instruction mnemonics for the assembler. The mnemonics are grouped according to the 6-bit hash value of the mnemonics - values range from 10H to 1FH.

For each instruction, there is a pointer to its codemacro definition.

*/

```

#include (:f1:mnem.lit)
= $nolist
#include (:f1:cmlink.ext)
= $nolist

```

/***** MNEMONICS TABLE *****/

4 1 declare

/*

* HASH VALUE (HEX) *

*/

jae	opcode3 public data (0,3,.jae1,"JAE"),	/* 10 */
jbe	opcode3 public data (0,3,.jbe1,"JBE"),	/* 11 */
jc	opcode2 public data (.jbe,2,.jg1,"JG"),	
lea	opcode3 public data (0,3,.leal,"LEA"),	/* 12 */
clc	opcode3 public data (.lea,3,.clcl,"CLC"),	
cmc	opcode3 public data (0,3,.cmcl,"CMC"),	/* 13 */
cld	opcode3 public data (.cmc,3,.cld1,"CLD"),	
iand	opcode3 public data (.cld,3,.and10,"AND"),	
loopz	opcode5 public data (0,5,.loopel,"LOOPZ"),	/* 14 */
aas	opcode3 public data (0,3,.aas1,"AAS"),	/* 15 */
jge	opcode3 public data (0,3,.jge1,"JGE"),	/* 16 */
jl	opcode2 public data (.jge,2,.jll,"JL"),	
sbb	opcode3 public data (0,3,.sbb11,"SBB"),	/* 17 */
in	opcode2 public data (.sbb,2,.in4,"IN"),	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

```

das      opcode3      data (0,3,.das1,"DAS"),      /* 18 */
cli      opcode3 public data (.das,3,.cli1,"CLI"),
jna      opcode3      data (0,3,.jnb1,"JNA"),      /* 19 */
jo       opcode2 public data (.jna,2,.jo1,"JO"),
jnb      opcode3      data (0,3,.jae1,"JNB"),      /* 1a */
jp       opcode2      data (.jnb,2,.jp1,"JP"),
neg      opcode3      data (.jp,3,.neg2,"NEG"),
inc      opcode3 public data (.neg,3,.inc3,"INC"),
jnc      opcode3      data (0,3,.jae1,"JNC"),      /* 1b */
esc      opcode3      data (.jnc,3,.esc3,"ESC"),
jle      opcode3      data (.esc,3,.jle1,"JLE"),
lahf     opcode4 public data (.jle,4,.lahf1,"LAHF"),
cbw      opcode3      data (0,3,.cbw1,"CBW"),      /* 1c */
MOVSW    opcode5      data (.cbw,5,.MOVSW1,"MOVSW"),
icall    opcode4 public data (.MOVSW,4,.call13,"CALL"),
js       opcode2      data (0,2,.js1,"JS"),      /* 1d */
jne      opcode3 public data (.js,3,.jne1,"JNE"),
jnae     opcode4      data (0,4,.jb1,"JNAE"),      /* 1e */
cwd      opcode3 public data (.jnae,3,.cwd1,"CWD"),
jpe      opcode3      data (0,3,.jpl1,"JPE"),      /* 1f */
jng      opcode3      data (.jpe,3,.jle1,"JNG"),
jnbe     opcode4 public data (.jng,4,.jal,"JNBE");

```

5 1 end mnem2;

MODULE INFORMATION:

```

CODE AREA SIZE      = 0000H      00
CONSTANT AREA SIZE  = 0122H      2900
VARIABLE AREA SIZE  = 0000H      00
MAXIMUM STACK SIZE  = 0000H      00
204 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE MNEM3

OBJECT MODULE PLACED IN MNEM3.OBJ

COMPILER INVOKED BY: :F0: MNEM3.PLH DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100)

```

1      $title ("INSTRUCTION MNEMONICS MODULE - PART 3")
      mnem3:
      do;

      /*

      modified 6/16/81 R. Silberstein

      */

      /***** INSTRUCTION MNEMONICS *****/
      /*

      This is all the instruction mnemonics for
      the assembler. The mnemonics are grouped
      according to the 6-bit hash value of the
      mnemonics - values range from 20H to 2FH.

      For each instruction, there is a pointer to
      its codemacro definition.

      */

      $include (:f1:mnem.lit)
=      $nolist
      $include (:f1:cmlink.ext)
=      $nolist

      /***** MNEMONICS TABLE *****/

4      1      declare

      /*

      * HASH VALUE (HEX) *
      -----

      */
      isal      opcod3      data (0,3,.sal4,"SAL"),      /* 20 */
      STOSW      OPCOD5      DATA (.ISAL,5,.STOSW1,"STOSW"),
      cmp        opcod3 public data (.STOSW,3,.cmp11,"CMP"),
      rcl        opcod3      data (0,3,.rcl4,"RCL"),      /* 21 */
      ior        opcod2 public data (.rcl,2,.or10,"OR"),
      loopnz     opcod6      data (0,6,.loopnz1,"LOOPNZ"), /* 22 */
      sahf       opcod4      data (.loopnz,4,.sahf1,"SAHF"),
      callf      opcod5 public data (.sahf,5,.callf2,"CALLF"),
      lds        opcod3      data (0,3,.lds1,"LDS"),      /* 23 */
      div        opcod3 public data (.lds,3,.div2,"DIV"),
      jnge       opcod4      data (0,4,.j11,"JNGE"),      /* 24 */
      jnl        opcod3      data (.jnge,3,.jel1,"JNL"),
      jz         opcod2      data (.jnl,2,.jel,"JZ"),
      les        opcod3 public data (.jz,3,.les1,"LES"),
      sar        opcod3 public data (0,3,.sar4,"SAR"),      /* 25 */
      /* 26 */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

jno      opcode3      data (0,3,.jno1,"JNO"),      /* 27 */
rcr      opcode3      data (.jno,3,.rcr4,"RCR"),
rep      opcode3      data (.rcr,3,.rep1,"REP"),
ishl     opcode3      data (.rep,3,.sal4,"SHL"),
jmp      opcode3 public data (.ishl,3,.jmp2,"JMP"),
jnp      opcode3      data (0,3,.jnp1,"JNP"),      /* 28 */
hlt      opcode3 public data (.jnp,3,.hlt1,"HLT"),
jnle     opcode4      data (0,4,.jgl1,"JNLE"),      /* 29 */
jpo      opcode3      data (.jnle,3,.jnp1,"JPO"),
lock     opcode4 public data (.jpo,4,.lock1,"LOCK"),
scas     opcode4      data (0,4,.scas2,"SCAS"),      /* 2a */
stc      opcode3      data (.scas,3,.stc1,"STC"),
sub      opcode3      data (.stc,3,.sub11,"SUB"),
xchg     opcode4 public data (.sub,4,.xchg6,"XCHG"),
jns      opcode3      data (0,3,.jns1,"JNS"),      /* 2b */
std      opcode3      data (.jns,3,.std1,"STD"),
int      opcode3      data (.std,3,.int2,"INT"),
ret      opcode3 public data (.int,3,.ret3,"RET"),
repe     opcode4      data (0,4,.rep1,"REPE"),      /* 2c */
SCASB    OPCOD5      DATA (.REPE,5,.SCASB1,"SCASB"),
idiv     opcode4 public data (.SCASB,4,.idiv2,"IDIV"),
nop      opcode3      data (0,3,.nop1,"NOP"),      /* 2d */
rol      opcode3      data (.nop,3,.rol4,"ROL"),
shr      opcode3      data (.rol,3,.shr4,"SHR"),
jmpf     opcode4 public data (.shr,4,.jmpf2,"JMPF"),
mul      opcode3 public data (0,3,.mul2,"MUL"),      /* 2e */
pop      opcode3 public data (0,3,.pop4,"POP");      /* 2f */

```

5 1 end mnem3;

MODULE INFORMATION:

```

CODE AREA SIZE      = 0000H      00
CONSTANT AREA SIZE  = 0160H      3520
VARIABLE AREA SIZE  = 0000H      00
MAXIMUM STACK SIZE  = 0000H      00
209 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE MNEM4

OBJECT MODULE PLACED IN MNEM4.OBJ

COMPILER INVOKED BY: :F0: MNEM4.PLM DEBJS OPTIMIZE(2) DATE(10/5/81) PAGESWIDTH(100)

```
1      $title ("INSTRUCTION MNEMONICS MODULE - PART 4")
```

```
      mnem4:
```

```
      do:
```

```
      /*
```

```
          modified 6/16/81 R. Silberstein
```

```
      */
```

```
      /***** INSTRUCTION MNEMONICS *****/
```

```
      /*
```

This is all the instruction mnemonics for the assembler. The mnemonics are grouped according to the 6-bit hash value of the mnemonics - values range from 30H to 3FH.

For each instruction, there is a pointer to its codemacro definition.

```
      /*
```

```
      $include (:f1:mnem.lit)
```

```
=      $nolist
```

```
      $include (:f1:cmlink.ext)
```

```
=      $nolist
```

```
      /***** MNEMONICS TABLE *****/
```

```
4      1      declare
```

```
      /*
```

```
          * HASH VALUE (HEX) *
```

```
      /*
      sti      opcode3 public data (0,3,.sti1,"STI"),      /* 30 */
      retf     opcode4      data (0,4,.retf3,"RETF"),      /* 31 */
      inot     opcode3 public data (.retf,3,.not2,"NOT"),
      lods     opcode4      data (0,4,.lods2,"LODS"),      /* 32 */
      jnz      opcode3      data (.lods,3,.jne1,"JNZ"),
      mov      opcode3 public data (.jnz,3,.mov17,"MOV"),
      ror      opcode3      data (0,3,.ror4,"ROR"),      /* 33 */
      cmps     opcode4 public data (.ror,4,.cmps2,"CMPS"),
      LODSB    OPCOD5      DATA (0,5,.LODSB1,"LODSB"),      /* 34 */
      iret     opcode4 public data (.lods,4,.iret1,"IRET"),
      wait     opcode4      data (0,4,.wait1,"WAIT"),      /* 35 */
      CMPSB    OPCOD5      DATA (.WAIT,5,.CMPSB1,"CMPSB"),
      popf     opcode4 public data (.CMPSB,4,.popf1,"POPF"),
                                     /* 36 */
      imul     opcode4 public data (0,4,.imul2,"IMUL"),      /* 37 */
      out      opcode3 public data (0,3,.out4,"OUT"),      /* 38 */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

retn    opcode4      data (0,4,.ret3,"RETN"),      /* 39 */
ixor    opcode3      data (.retn,3,.xor10,"XOR"),
xlat    opcode4 public data (.ixor,4,.xlat1,"XLAT"),
repne   opcode5      data (0,5,.repne1,"REPNE"),    /* 3a */
into    opcode4      data (.repne,4,.into1,"INT0"),
loop    opcode4      data (.into,4,.loop1,"LOOP"),
jmpls   opcode4 public data (.loop,4,.jmps1,"JMP5"),

/* 3b */
/* 3c */
/* 3d */
/* 3e */
/* 3f */

jcxz    opcode4      data (0,4,.jcxz1,"JCXZ"),
loope   opcode5 public data (.jcxz,5,.loope1,"LOOPE");

```

/***** END OF MNEMONIC TABLE *****/

5 1 end mhem4;

MODULE INFORMATION:

```

CODE AREA SIZE      = 0000H      00
CONSTANT AREA SIZE  = 0005H      213D
VARIABLE AREA SIZE  = 0000H      00
MAXIMUM STACK SIZE  = 0000H      00
197 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-II PL/M-86 V2.0 COMPTLATION OF MODULE SYMB

OBJECT MODULE PLACED IN SYMB.OBJ

COMPILER INVOKED BY: :FO: SYMB.PLN DEBUG OPTIMIZE(2) DATE(10/5/81) PAGESWIDTH(100) XPT:

```

1      $title ("SYMBOL TABLE MODULE")

```

```

      symb:

```

```

      do:

```

```

/*

```

```

      modified 3/25/81 R. Silberstein
      modified 3/28/81 R. Silberstein
      modified 3/30/81 R. Silberstein
      modified 4/15/81 R. Silberstein
      modified 4/16/81 R. Silberstein
      modified 4/20/81 R. Silberstein
      modified 6/16/81 R. Silberstein
      modified 7/24/81 R. Silberstein

```

```

*/

```

```

/*

```

```

      This is the module to perform all symbol table
      handling. There are 2 different kinds of symbols,
      codemacro mnemonics and user defined symbols.

```

```

      The codemacro symbols are entered into the
      symbol table through the hash vector "CODEMACROENTRY",
      whereas the user symbols uses the hash vector
      "SYMBENTRY". Each symbol enters the symbol table through
      hash vector element "i", where i is the hash function of
      the symbol. The function is defined as:

```

$$H(S) = (C_1 + C_2 + \dots + C_i + \dots + C_n) \bmod 64$$

```

      where Ci is the ascii code of the i'th symbolcharacter.

```

```

*/

```

```

      $include (:f1:macro.lit)

```

```

=      $nolist

```

```

      $include (:f1:equals.lit)

```

```

=      $nolist

```

```

      $include (:f1:struc.lit)

```

```

=      $nolist

```

```

      $INCLUDE (:F1:DEV.LIT)

```

```

=      $nolist

```

```

      $include (:f1:mnem.ext)

```

```

=      $nolist

```

```

      $include (:f1:subr1.ext)

```

```

=      $nolist

```

```

      $include (:f1:subr2.ext)

```

```

=      $nolist

```

```

      $include (:f1:files.ext)

```

```

=      $nolist

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
      $INCLUDE (:F1:TEXT.EXT)
=     $nolist
      $INCLUDE (:F1:IO.EXT)
=     $nolist
152 1  DECLARE SYMBOLDEVICE BYTE EXTERNAL;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$object

/* Global variables: */

153 1

```
dcl
codemacroentry (64) addr      /* opcode mnemonic entry */
data(
    .push,.repz,0,.aaa,0,.movs,.pushf,.MOVSB,
    .adc,.add,.CMPSW,.ja,.dec,.loopne,0,.repnz,
    .jae,.jg,.clc,.land,.loopz,.aas,.jl,.in,
    .cli,.jo,.inc,.lahf,.icall,.jne,.cni,.jnbe,
    .cmp,.ior,.callf,.div,.les,0,.sar,.jmp,
    .hlt,.lock,.xchg,.ret,.idiv,.jmpf,.mul,.pop,
    .sti,.inot,.mov,.cmps,.iret,.popf,0,.imul,
    .out,.xlat,.jmps,0,0,0,0,.loope),

symentry (64) addr,          /* user symbol entry */
symtab(1) byte at (.memory), /* symbol table */
freeptr      addr public,    /* adr. of next free symb. tab byte */
endofsymtab   addr PUBLIC,   /* last symbol table byte */
symbolheadlg  byte,          /* length of head for each symbol */
attributelg  byte,          /* length of symbol attributes */
codemacheadlg byte,          /* length of codemacrohead */
overflowlimit addr,         /* used to test overflow */
col           byte;         /* current column position */
```

154 1

```
DECLARE
ALPHAROOT ADDRESS,          /* ROOT OF ALPHABETIZED SYMBOL LIST */
ALPHASYMPTR ADDRESS,       /* POINTER TO CURRENT SYMBOL IN ALPHA LIST */
ALPHASYM BASED ALPHASYMPTR SYMBOLHEAD, /* SYMBOL TEMPLATE */
SORTSYMPTR ADDRESS,       /* POINTER TO SYMBOL BEING INSERTED */
SORTSYM BASED SORTSYMPTR SYMBOLHEAD; /* SYMBOL TEMPLATE */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$select
/***** SUBROUTINES *****/
```

```
155 1  outbyt: proc (ch); /* print one byte to symbol file */
156 2      dcl ch byte;
157 2      if ch=cr then$do /* update column position */
159 3          col=0;
160 3      else$do
162 3          if ch <> lf then col=col+1;
            end$if;
165 2      call outsymbolbyte(ch);
166 2  end outbyt;

167 1  hash: proc (lg,pt) byte;
168 2      dcl (lg,lg,h) byte,pt addr,asc based pt (1) byte;
169 2      i=0fff; h=0;
171 2      do while (i:=i+1) < lg;
172 3          h=h+asc(i);
173 3      end$while;
174 2      return h and 3fh;
175 2  end hash;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

```
/* search for either a user symbol or a codemacro */
```

```
176 1  search: proc(lg,pt1,pt2,pt3,headlg) byte;
177 2      dcl (lg,headlg) byte,(pt1,pt2,pt3) addr,
            ascii based pt1(1) byte,symbptr based pt2 addr,
            entrytab based pt3 (64) addr,
            currentpt addr,next based currentpt addr,
            symbhead based currentpt symbolhead;

178 2      currentpt=entrytab(hash(lg,.ascii(0)));
179 2      do while currentpt <> 0;
180 3          if lg = symbhead.slength then$do
182 4              if equal(lg,currentpt+headlg,.ascii(0)) then$do
184 5                  symbptr=currentpt+3;
185 5                  return true;
186 5              end$if;
            end$if;
            currentpt=next;
188 3          end$while;
189 3          return false;
190 2      return false;
191 2  end search;

/* enter either new symbol or new codemacro */
192 1  new: proc(lg,pt1,pt2,headlg,pt3) byte;
193 2      dcl (lg,headlg) byte,(pt1,pt2,pt3) addr,
            ascii based pt1 (1) byte,entrytab based pt2 (64) addr,
            symptr based pt3 addr,
            current addr,currentcontent based current addr,
            symb based freeptr symbolhead;
194 2      if freeptr > overflowlimit - (lg+headlg) then$DO
196 3          CALL OUTTEXT (.SYMBFULLERRTEXT);
197 3          CALL SYSTEM$RESET;
198 3      END$IF;
199 2      current=.entrytab(hash(lg,.ascii(0)));
200 2      SYMB.NEXT = CURRENTCONTENT;
```

```
201 2      currentcontent=freept;
202 2      symptr=freept+3;
203 2      symb.slength=lg;
204 2      call copy(lg,.ascii(0),freept+headlg);
205 2      freept=freept+headlg+lg;
206 2      return true;
207 2      end new;

208 1      newsymbol: proc (lg,asciiptr,returnpt) byte public;
209 2      dcl lg byte,(asciiptr,returnpt) addr;
210 2      return new(lg,asciiptr,.symbentry,symbolheadlg,returnpt);
211 2      end newsymbol;

212 1      newmacro: proc (lg,asciiptr,codemacdefpt) byte public;
213 2      dcl lg byte,(asciiptr,codemacdefpt,retpt) addr,
          cmaddr based retpt addr;
214 2      if new(lg,asciiptr,.codemacroentry,codemacheadlg,.retpt) then$do
216 3      cmaddr=codemacdefpt;
217 3      return true;
218 3      end$if;
219 2      return false;
220 2      end newmacro;

221 1      findsymbol: proc(lg,stradr,result) byte public;
222 2      dcl lg byte,(stradr,result) addr;
223 2      return search(lg,stradr,result,.symbentry(0),symbolheadlg);
224 2      end findsymbol;

225 1      getattributes: proc(symbadr,dest) public;
226 2      dcl (symbadr,dest) addr,symb based symbadr symbolstruc;
227 2      call copy(attributelg,.symb.slength,dest);
228 2      end getattributes;

229 1      enterattributes: proc(symbadr,source) public;
230 2      dcl (symbadr,source) addr,symb based symbadr symbolstruc;
231 2      call copy(attributelg,source,.symb.slength);
232 2      end enterattributes;

233 1      findcodemacro: proc(lg,stradr,result) byte public;
234 2      dcl lg byte,(stradr,result) addr;
235 2      return search(lg,stradr,result,.codemacroentry(0),codemacheadlg);
236 2      end findcodemacro;

237 1      new$cm$body: PROC (lg,ptr) byte public;
238 2      dcl lg byte,ptr addr;
239 2      if freept > overflowlimit-lg then return false;
241 2      call copy (lg,ptr,freept);
242 2      freept=freept+lg;
243 2      return true;
244 2      end$proc new$cm$body;

245 1      new$cm$name: PROC (lg,asciiptr,returnptr) byte public;
246 2      dcl lg byte,(asciiptr,returnptr) addr;
247 2      return new(lg,asciiptr,.codemacroentry,5,returnptr);
248 2      end$proc new$cm$name;

249 1      SORTSYMBOLS: PROCEDURE;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

```
250 2 DECLARE
      CURRENT ADDRESS,
      CURRENTCONTENT BASED CURRENT ADDRESS,
      NEXT ADDRESS,
      I BYTE;

251 2 ALPHALOCFOUND: PROCEDURE BYTE;
252 3 DECLARE
      SORTNAMEPTR ADDRESS,
      SORTNAME BASED SORTNAMEPTR (1) BYTE,
      ALPHANAMEPTR ADDRESS,
      ALPHANAME BASED ALPHANAMEPTR (1) BYTE,
      I BYTE;

253 3 SORTNAMEPTR = SORTSYMPTR + SYMBOLHEADLG;
254 3 ALPHANAMEPTR = ALPHASYMPTR + SYMBOLHEADLG;
255 3 DO I = 1 TO SORTSYN.LENGTH;
256 4 IF I > ALPHASYN.LENGTH THEN RETURN FALSE;
258 4 IF SORTNAME (I-1) > ALPHANAME (I-1) THEN RETURN FALSE;
260 4 IF SORTNAME (I-1) < ALPHANAME (I-1) THEN RETURN TRUE;
262 4 END;
263 3 RETURN TRUE;
264 3 END ALPHALOCFOUND;

265 2 FIXCHAIN: PROCEDURE;
266 3 SORTSYN.NEXT = ALPHASYMPTR;
267 3 CURRENTCONTENT = .SORTSYN;
268 3 END FIXCHAIN;

269 2 INSERTALPHA: PROCEDURE;
270 3 CURRENT, ALPHASYMPTR = .ALPHAROOT;
271 3 DO WHILE (ALPHASYMPTR := ALPHASYN.NEXT) <> 0;
272 4 IF ALPHALOCFOUND THEN$DO
274 5 CALL FIXCHAIN;
275 5 RETURN;
276 5 END$IF;
277 4 CURRENT = ALPHASYMPTR;
278 4 END$WHILE;
279 3 CALL FIXCHAIN;
280 3 END INSERTALPHA;

281 2 ALPHAROOT = 0;
282 2 DO I = 0 TO LAST (SYMBENTRY);
283 3 SORTSYMPTR = SYMBENTRY (I);
284 3 DO WHILE SORTSYMPTR <> 0;
285 4 NEXT = SORTSYN.NEXT;
286 4 CALL INSERTALPHA;
287 4 SORTSYMPTR = NEXT;
288 4 END$WHILE;
289 3 END;
290 2 END SORTSYMBOLS;

291 1 outerlf: proc;
292 2 call outbyt(cr);
293 2 call outbyt(lf);
294 2 end outerlf;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

295 1  printsymbols: proc(typ);      /* print all symbols to file */
296 2      dcl (typ,i) byte;

297 2      advancetonext: proc; /* advance to next column (16,32,48,64) */
298 3      dcl (x,y) byte;
299 3      x=col/16;
300 3      y=16-(col mod 16);
301 3      if x >= 4 then$do
303 4          call outcrif;
304 4      else$do
306 4          IF SYMBOL$DEVICE = PRINTER THEN$DO
308 5              DO WHILE (Y := Y - 1) <> 255;
309 6                  CALL OUTBYT (SPACE);
310 6              END$WHILE;
311 5          ELSE$DO
313 5              col=col+y;
314 5              call outsymbolbyte(tab);
315 5              if y > 8 then call outsymbolbyte(tab);
318 4          end$if;
319 3      end advancetonext;

320 2  OUTBYTES: proc(lg,p);
321 3      dcl lg byte,p addr,asc based p byte;
322 3      DO WHILE (LG := LG - 1) <> 0FFH;
323 4          call outbyt(asc);
324 4          P = P + 1;
325 4      end$while;
326 3      end OUTBYTES;

327 2  printsymb: proc(p); /* print single symbol */
328 3      dcl p addr,x based p symbolhead,
329 3          a addr,ascii based a (1) byte,help(4) byte;
330 3      a=p+symbolheadlg;
331 3      if X.SIYPE=typ then$do
332 4          if col+x.slength+5 > 79 then call outcrif;
333 4          call hex2out(x.offset,.help(0));
334 4          CALL OUTBYTES (4, .HELP);
335 4          call outbyt(space);
336 4          CALL OUTBYTES (X.SLENGTH, .ASCII);
337 4          call advancetonext;
338 4      end$if;
339 3      end printsymb;

/* print symbols main program */

341 2      col=0;
342 2      CALL OUTBYTES (4, .("0000")); /* print header */
343 2      call outbyt(space);
344 2      if typ=lab then call OUTBYTES(6,.( "LABELS"));
345 2      if typ=variable then call OUTBYTES(9,.( "VARIABLES"));
346 2      if typ=number then call OUTBYTES(7,.( "NUMBERS"));
347 2      call outcrif;
348 2      ALPHASYMPTR = ALPHAROOT;
349 2      DO WHILE ALPHASYMPTR <> 0;
350 3          CALL PRINTSYMB (ALPHASYMPTR);
351 3          ALPHASYMPTR = ALPHASYM.NEXT;

```

CCP/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-704

```
355 3      END;
356 2      if col <> 0 then call outcrLf;

358 2      end printsymbols;

359 1      symbterminate: proc public; /* print symbol table */
360 2      if SYMDEV = NULL THEN RETURN; /* no need to sort, etc. */
362 2      CALL SORTSYMBOLS;
363 2      CALL OUTSYMBOLBYTE (F0xMFEED);
364 2      call printsymbols(variable); /* variables */
365 2      CALL OUTCRLF;
366 2      call printsymbols(number); /* numbers */
367 2      CALL OUTCRLF;
368 2      call printsymbols(lab); /* labels */
369 2      end symbterminate;

370 1      symbinit: proc public;
371 2      dcl i byte;
372 2      dcl symb symbolhead at (.i), codm codemacrohead at (.i);
373 2      end$of$symtab=(endbuf/256)*256-1;
374 2      freep=.symtab(0);
375 2      CALL FILL (0, SIZE (SYMBENTRY), .SYMBENTRY);
376 2      symbolheadlg=.symb.baseindex-.symb.next+1;
377 2      attributelg=symbolheadlg-3;
378 2      codemacheadlg=.codm.defptr-.codm.next+2;
379 2      overflowlimit=end$of$symtab-symbolheadlg;
380 2      end symbinit;

381 1      end$module symb;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
328	009CH	2	A. WORD 328 329 337
10	0000H	1	AAA. BYTE EXTERNAL(2) 153
10	0000H	1	AAS. BYTE EXTERNAL(18) 153
10	0000H	1	ADC. BYTE EXTERNAL(6) 153
10	0000H	1	ADD. BYTE EXTERNAL(7) 153
2			ADDR. LITERALLY 12 15 18 21 27 45 48
			51 54 92 153 154 168 177 193 209 213
			222 226 230 234 238 246 321 323 372
297	03D9H	100	ADVANCETONEXT. PROCEDURE STACK=0010H 338
251	0290H	114	ALPHALOCFOUND. PROCEDURE BYTE STACK=0002H 272
252	0000H	1	ALPHANAME. BYTE BASED(ALPHANAMEPTR) ARRAY(1) 258 260
252	0098H	2	ALPHANAMEPTR. WORD 252 254 258 260
35	0000H		ALPHANUMERIC. PROCEDURE BYTE EXTERNAL(63) STACK=0000H
154	0086H	2	ALPHAROOT. WORD 270 281 351
154	0000H	12	ALPHASYM. STRUCTURE BASED(ALPHASYMPTR) 256 271 354
154	0088H	2	ALPHASYMPTR. WORD 154 254 256 266 270 271 277 351
			352 353 354
321	0000H	1	ASC. BYTE BASED(P) 323
168	0000H	1	ASC. BYTE BASED(PT) ARRAY(1) 172
328	0000H	1	ASCTI. BYTE BASED(A) ARRAY(1) 337
193	0000H	1	ASCTI. BYTE BASED(PT1) ARRAY(1) 199 204
177	0000H	1	ASCTI. BYTE BASED(PT1) ARRAY(1) 178 182
38	0000H		ASCTICHAR. PROCEDURE BYTE EXTERNAL(64) STACK=0000H
212	0006H	2	ASCTIPTR. WORD PARAMETER AUTOMATIC 213 214
245	0006H	2	ASCTIPTR. WORD PARAMETER AUTOMATIC 246 247
208	0006H	2	ASCTIPTR. WORD PARAMETER AUTOMATIC 209 210
96	0000H	1	ASM86TEXT. BYTE ARRAY(1) EXTERNAL(94)
153	009FH	1	ATTRIBUTELG. BYTE 227 231 377
154	0008H	1	BASEINDEX. BYTE MEMBER(ALPHASYM)
154	0008H	1	BASEINDEX. BYTE MEMBER(SORTSYM)
193	0008H	1	BASEINDEX. BYTE MEMBER(SYMB)
372	0008H	1	BASEINDEX. BYTE MEMBER(SYMB) 376
226	0008H	1	BASEINDEX. BYTE MEMBER(SYMB)
328	0008H	1	BASEINDEX. BYTE MEMBER(X)
230	0008H	1	BASEINDEX. BYTE MEMBER(SYMB)
177	0008H	1	BASEINDEX. BYTE MEMBER(SYMBHEAD)
8			BASEREGBIT. LITERALLY
8			BASEREGCOUNT. LITERALLY
8			BREGBIT. LITERALLY
8			BREGCOUNT. LITERALLY
4			BYT. LITERALLY
10	0000H	1	CALLF. BYTE EXTERNAL(31) 153
59	0000H	1	CH. BYTE PARAMETER 60
32	0000H	1	CH. BYTE PARAMETER 33
26	0000H	1	CH. BYTE PARAMETER 27
38	0000H	1	CH. BYTE PARAMETER 39
29	0000H	1	CH. BYTE PARAMETER 30
155	0004H	1	CH. BYTE PARAMETER AUTOMATIC 156 157 162 165
41	0000H	1	CH. BYTE PARAMETER 42

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

35	0000H	1	CH	BYTE PARAMETER	35
56	0000H	1	CH	BYTE PARAMETER	57
62	0000H	1	CH	BYTE PARAMETER	63
111	0000H	1	CHAR	BYTE PARAMETER	112
114	0000H	1	CHAR	BYTE PARAMETER	115
10	0000H	1	CLC	BYTE EXTERNAL(15)	153
10	0000H	1	CLI	BYTE EXTERNAL(21)	153
96	0000H	1	CLOSEERRTEXT	BYTE ARRAY(1) EXTERNAL(100)	
130	0000H		CLOSEFILE	PROCEDURE BYTE EXTERNAL(120) STACK=0000H	
85	0000H		CLOSEHEX	PROCEDURE EXTERNAL(83) STACK=0000H	
83	0000H		CLOSEINCLUDE	PROCEDURE EXTERNAL(82) STACK=0000H	
87	0000H		CLOSEPKINT	PROCEDURE EXTERNAL(84) STACK=0000H	
79	0000H		CLOSESOURCE	PROCEDURE EXTERNAL(80) STACK=0000H	
89	0000H		CLOSESYMBOL	PROCEDURE EXTERNAL(85) STACK=0000H	
213	0000H	2	CMADDR	WORD BASED(CREPT)	216
10	0000H	1	CMP	BYTE EXTERNAL(29)	153
10	0000H	1	CMPS	BYTE EXTERNAL(47)	153
10	0000H	1	CMPSW	BYTE EXTERNAL(8)	153
3			CODE	LITERALLY	
153	00A0H	1	CODEHACHEADLG	BYTE	214 235 378
153	0000H	128	CODEMACROENTRY	WORD ARRAY(64) DATA	214 235 247
8			CODEMACROHLAD	LITERALLY	372
372	00A0H	5	COOM	STRUCTURE AT	378
212	0004H	2	COOMACDEFPT	WORD PARAMETER AUTOMATIC	213 216
153	00A1H	1	COL	BYTE	159 163 299 300 313 332 341 356
9			CONSOLE	LITERALLY	
117	0000H		CONSTAT	PROCEDURE BYTE EXTERNAL(115) STACK=0000H	
17	0000H		COPY	PROCEDURE EXTERNAL(57) STACK=0000H	204 227
				231 241	
2			CR	LITERALLY	157 292
142	0000H		CREATEFILE	PROCEDURE BYTE EXTERNAL(124) STACK=0000H	
150	0000H		CRLF	PROCEDURE EXTERNAL(127) STACK=0000H	
250	0092H	2	CURRENT	WORD	250 267 270 277
193	008EH	2	CURRENT	WORD	193 199 200 201
193	0000H	2	CURRENTCONTENT	WORD BASED(CURRENT)	200 201
250	0000H	2	CURRENTCONTENT	WORD BASED(CURRENT)	267
177	008CH	2	CURRENTPT	WORD	177 178 179 180 182 184 188
10	0000H	1	CWD	BYTE EXTERNAL(27)	153
44	0000H	2	D	WORD PARAMETER	45
47	0000H	2	D	WORD PARAMETER	48
50	0000H	2	D	WORD PARAMETER	51
53	0000H	2	D	WORD PARAMETER	54
17	0000H	2	D	WORD PARAMETER	18
2			DCL	LITERALLY	
10	0000H	1	DEC	BYTE EXTERNAL(10)	153
53	0000H		DECOUT	PROCEDURE EXTERNAL(69) STACK=0000H	
372	0003H	2	DEFPTR	WORD MEMBER(COOM)	378
3			DELETEDSYM	LITERALLY	
133	0000H		DELETEFILE	PROCEDURE EXTERNAL(121) STACK=0000H	
225	0004H	2	DEST	WORD PARAMETER AUTOMATIC	226 227
91	0000H	1	DEV	BYTE PARAMETER	92
29	0000H		DIGIT	PROCEDURE BYTE EXTERNAL(61) STACK=0000H	
121	0000H	1	DISKNUMBER	BYTE PARAMETER	122
96	0000H	1	DISKREADERRTEXT	BYTE ARRAY(1) EXTERNAL(97)	
96	0000H	1	DISKWRITEERRTXT	BYTE ARRAY(1) EXTERNAL(98)	
10	0000H	1	DIV	BYTE EXTERNAL(32)	153
147	0000H	2	DMAADDRESS	WORD PARAMETER	148

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

3		DOUBLEDEFINED. . .	LITERALLY	
4		DWRD	LITERALLY	
2		ELSEDD	LITERALLY	
106	0000H	2 ENDRUF	WORD EXTERNAL(110)	373
2		ENDCASE	LITERALLY	
2		ENDDD	LITERALLY	
2		ENDFOREVER	LITERALLY	
2		ENDIF	LITERALLY	163 186 316
2		ENDMODULE	LITERALLY	
2		ENDOFFILE	LITERALLY	
153	0082H	2 ENDOFSYMTAB	WORD PUBLIC	373 379
2		ENDPROC	LITERALLY	
96	0000H	1 ENDTXT	BYTE ARRAY(1) EXTERNAL(92)	
2		ENDWHILE	LITERALLY	
229	01CCH	23 ENTERATTRIBUTES. . .	PROCEDURE PUBLIC STACK=000CH	
177	0000H	128 ENTRYTAB	WORD BASED(P13) ARRAY(64)	178
193	0000H	128 ENTRYTAB	WORD BASED(P12) ARRAY(64)	199
44	0000H	EQUAL	PROCEDURE BYTE EXTERNAL(66) STACK=0000H	182
3		ERROR	LITERALLY	
2		FALSE	LITERALLY	190 219 240 257 259
103	0000H	1 FCB	BYTE ARRAY(1) EXTERNAL(107)	
104	0000H	1 FCB16	BYTE ARRAY(1) EXTERNAL(108)	
136	0000H	2 FCBADDRESS	WORD PARAMETER	137
139	0000H	2 FCBADDRESS	WORD PARAMETER	140
142	0000H	2 FCBADDRESS	WORD PARAMETER	143
127	0000H	2 FCBADDRESS	WORD PARAMETER	128
130	0000H	2 FCBADDRESS	WORD PARAMETER	131
124	0000H	2 FCBADDRESS	WORD PARAMETER	125
133	0000H	2 FCBADDRESS	WORD PARAMETER	134
23	0000H	2 FCBADR	WORD PARAMETER	24
23	0000H	FILEABORT	PROCEDURE EXTERNAL(59) STACK=0000H	
8		FILESTRUCTURE	LITERALLY	
8		FILESTRUCTURE	LITERALLY	
94	0000H	FILESETUP	PROCEDURE BYTE EXTERNAL(87) STACK=0000H	
26	0000H	FILL	PROCEDURE EXTERNAL(60) STACK=0000H	375
91	0000H	2 FILNAM	WORD PARAMETER	92
91	0000H	2 FILTYP	WORD PARAMETER	92
233	01E3H	27 FINDCODEMACRO	PROCEDURE BYTE PUBLIC STACK=001EH	
221	019AH	27 FINDSYMBOL	PROCEDURE BYTE PUBLIC STACK=001EH	
265	0302H	26 FIXCHAIN	PROCEDURE STACK=0002H	274 279
2		FOREVER	LITERALLY	
2		FORMFEED	LITERALLY	363
153	0080H	2 FREEPT	WORD PUBLIC	193 194 200 201 202 203
			204 205 239 241 242 374	
100	0000H	1 FUNC	BYTE PARAMETER	101
97	0000H	1 FUNC	BYTE PARAMETER	98
225	01B5H	23 GETATTRIBUTES. . . .	PROCEDURE PUBLIC STACK=000CH	
168	00A3H	1 H	BYTE	170 172 174
167	0024H	50 HASH	PROCEDURE BYTE STACK=0006H	178 199
192	0006H	1 HEADLG	BYTE PARAMETER AUTOMATIC	193 194 204 205
176	0004H	1 HEADLG	BYTE PARAMETER AUTOMATIC	177 182
328	00A9H	4 HELP	BYTE ARRAY(4)	334 335
47	0000H	HEX1OUT	PROCEDURE EXTERNAL(67) STACK=0000H	
50	0000H	HEX2OUT	PROCEDURE EXTERNAL(68) STACK=0000H	334
10	0000H	1 HLT	BYTE EXTERNAL(36)	153
252	00A5H	1 I	BYTE	255 256 258 260
168	00A2H	1 I	BYTE	169 171 172

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

296	00A6H	1	I	BYTE			
250	00A4H	1	I	BYTE	232	283	
371	00ADH	1	I	BYTE	372		
10	0000H	1	IAND	BYTE EXTERNAL(16)		153	
10	0000H	1	ICALL	BYTE EXTERNAL(25)		153	
3			IDENT	LITERALLY			
10	0000H	1	IDIV	BYTE EXTERNAL(40)		153	
91	0000H		IFILESETUP	PROCEDURE EXTERNAL(86) STACK=0000H			
10	0000H	1	IMUL	BYTE EXTERNAL(50)		153	
10	0000H	1	IN	BYTE EXTERNAL(20)		153	
10	0000H	1	INC	BYTE EXTERNAL(23)		153	
8			INDEXRLGBIT	LITERALLY			
8			INDEXREGCOUNT	LITERALLY			
97	0000H	2	INFO	WORD PARAMETER		98	
100	0000H	2	INFO	WORD PARAMETER		101	
67	0000H		ININCLUDEBYTE	PROCEDURE BYTE EXTERNAL(74) STACK=0000H			
2			INIT	LITERALLY			
96	0000H	1	INITIALS	BYTE ARRAY(1) EXTERNAL(90)			
10	0000H	1	INOT	BYTE EXTERNAL(45)		153	
269	031CH	48	INSERTALPHA	PROCEDURE STACK=0006H		286	
65	0000H		INSOURCEBYTE	PROCEDURE BYTE EXTERNAL(73) STACK=0000H			
145	0000H		INTERROGATEDISK	PROCEDURE BYTE EXTERNAL(125) STACK=0000H			
10	0000H	1	IOR	BYTE EXTERNAL(30)		153	
8			IREGBIT	LITERALLY			
8			IREGCOUNT	LITERALLY			
10	0000H	1	IRET	BYTE EXTERNAL(48)		153	
10	0000H	1	JA	BYTE EXTERNAL(9)		153	
10	0000H	1	JAE	BYTE EXTERNAL(13)		153	
10	0000H	1	JG	BYTE EXTERNAL(14)		153	
10	0000H	1	JL	BYTE EXTERNAL(19)		153	
10	0000H	1	JMP	BYTE EXTERNAL(35)		153	
10	0000H	1	JMPF	BYTE EXTERNAL(41)		153	
10	0000H	1	JMPS	BYTE EXTERNAL(53)		153	
10	0000H	1	JNBE	BYTE EXTERNAL(28)		153	
10	0000H	1	JNE	BYTE EXTERNAL(26)		153	
10	0000H	1	JO	BYTE EXTERNAL(22)		153	
3			LAB	LITERALLY	344	368	
10	0000H	1	L&HF	BYTE EXTERNAL(24)		153	
			LAST	BUILTIN		282	
7			LEFTBRACKET	LITERALLY			
226	0000H	2	LENGTH	WORD MEMBER(SYMB)		227	
177	0003H	2	LENGTH	WORD MEMBER(SYMBHEAD)			
154	0003H	2	LENGTH	WORD MEMBER(SORTSYMB)			
154	0003H	2	LENGTH	WORD MEMBER(ALPHASYMB)			
230	0000H	2	LENGTH	WORD MEMBER(SYMB)		231	
193	0003H	2	LENGTH	WORD MEMBER(SYMB)			
372	0003H	2	LENGTH	WORD MEMBER(SYMB)			
328	0003H	2	LENGTH	WORD MEMBER(X)			
10	0000H	1	LES	BYTE EXTERNAL(33)		153	
32	0000H		LETTER	PROCEDURE BYTE EXTERNAL(62) STACK=0000H			
2			LF	LITERALLY	162	293	
245	0008H	1	LG	BYTE PARAMETER AUTOMATIC	246	247	
233	0008H	1	LG	BYTE PARAMETER AUTOMATIC	234	235	
208	0008H	1	LG	BYTE PARAMETER AUTOMATIC	209	210	
176	000CH	1	LG	BYTE PARAMETER AUTOMATIC	177	178	180 182
221	0008H	1	LG	BYTE PARAMETER AUTOMATIC	222	223	
237	0006H	1	LG	BYTE PARAMETER AUTOMATIC	238	239	241 242

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

167	0006H	1	LG	BYTE PARAMETER AUTOMATIC	168	171	
192	000CH	1	LG	BYTE PARAMETER AUTOMATIC	193	194	199 203
				204 205			
212	0008H	1	LG	BYTE PARAMETER AUTOMATIC	213	214	
320	0006H	1	LG	BYTE PARAMETER AUTOMATIC	321	322	
2			LIT.	LITERALLY 2 3 4 5 6 7 8			
				9			
10	0000H	1	LOCK	BYTE EXTERNAL(37)	153		
10	0000H	1	LOUPE	BYTE EXTERNAL(54)	153		
10	0000H	1	LOOPNE	BYTE EXTERNAL(11)	153		
10	0000H	1	LOUPE	BYTE EXTERNAL(17)	153		
96	0000H	1	MAKEERRTEXT	BYTE ARRAY(1) EXTERNAL(99)			
	0000H		MEMORY	BYTE ARRAY(0)	153		
97	0000H		MON1	PROCEDURE EXTERNAL(105) STACK=0000H			
100	0000H		MON2	PROCEDURE BYTE EXTERNAL(106) STACK=0000H			
10	0000H	1	MOV	BYTE EXTERNAL(46)	153		
10	0000H	1	MOVS	BYTE EXTERNAL(3)	153		
10	0000H	1	MOVSB	BYTE EXTERNAL(5)	153		
10	0000H	1	MUL	BYTE EXTERNAL(42)	153		
53	0000H	2	N	WORD PARAMETER	54		
50	0000H	2	N	WORD PARAMETER	51		
47	0000H	1	N	BYTE PARAMETER	48		
44	0000H	1	N	BYTE PARAMETER	45		
26	0000H	1	N	BYTE PARAMETER	27		
14	0000H	2	N	WORD PARAMETER	15		
17	0000H	1	N	BYTE PARAMETER	18		
3			NEGLECTED	LITERALLY			
192	00B7H	153	NEW	PROCEDURE BYTE STACK=0014H	210	214	247
237	01FEH	52	NEWCMDBODY	PROCEDURE BYTE PUBLIC STACK=000EH			
245	0232H	26	NEWCMNAME	PROCEDURE BYTE PUBLIC STACK=001EH			
212	016BH	47	NEWMACRO	PROCEDURE BYTE PUBLIC STACK=001EH			
208	0150H	27	NEWSYMBOL	PROCEDURE BYTE PUBLIC STACK=001EH			
250	0094H	2	NEXT	WORD 285 287			
193	0000H	2	NEXT	WORD MEMBER(SYMB)	200		
177	0000H	2	NEXT	WORD BASED(CURRENTPT)	138		
154	0000H	2	NEXT	WORD MEMBER(SORTSYM)	266	285	
154	0000H	2	NEXT	WORD MEMBER(CALPHASYM)	271	354	
372	0000H	2	NEXT	WORD MEMBER(SYMB)	376		
372	0000H	2	NEXT	WORD MEMBER(COON)	378		
177	0000H	2	NEXT	WORD MEMBER(SYMBHEAD)			
328	0000H	2	NEXT	WORD MEMBER(X)			
4			NIL	LITERALLY			
96	0000H	1	NO	BYTE EXTERNAL(89)			
8			NOOVERRIDEIT	LITERALLY			
8			NOOVERRIDEIT	LITERALLY			
9			NULL	LITERALLY 360			
3			NUMBER	LITERALLY 348 366			
7			OAND	LITERALLY			
7			OEQ	LITERALLY			
226	0006H	2	OFFSET	WORD MEMBER(SYMB)			
193	0009H	2	OFFSET	WORD MEMBER(SYMB)			
177	0009H	2	OFFSET	WORD MEMBER(SYMBHEAD)			
372	0009H	2	OFFSET	WORD MEMBER(SYMB)			
154	0009H	2	OFFSET	WORD MEMBER(SORTSYM)			
154	0009H	2	OFFSET	WORD MEMBER(CALPHASYM)			
230	0006H	2	OFFSET	WORD MEMBER(SYMB)			
328	0009H	2	OFFSET	WORD MEMBER(X)	334		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

7		OGE.	LITERALLY				
7		OGI.	LITERALLY				
7		OLAST.	LITERALLY				
7		OLE.	LITERALLY				
7		OLENGTH.	LITERALLY				
7		OLT.	LITERALLY				
7		OMOD.	LITERALLY				
7		ONE.	LITERALLY				
7		ONOT.	LITERALLY				
7		OFFSET.	LITERALLY				
7		OR.	LITERALLY				
96	0000H	1 OPENERTEXT.	BYTE ARRAY(1) EXTERNAL(95)				
124	0000H	OPENFILE.	PROCEDURE BYTE EXTERNAL(118) STACK=0000H				
73	0000H	OPENHEX.	PROCEDURE EXTERNAL(77) STACK=0000H				
71	0000H	OPENINCLUDE.	PROCEDURE EXTERNAL(76) STACK=0000H				
75	0000H	OPENPRINT.	PROCEDURE EXTERNAL(78) STACK=0000H				
127	0000H	OPENPROFILE.	PROCEDURE BYTE EXTERNAL(119) STACK=0000H				
69	0000H	OPENSOURCE.	PROCEDURE EXTERNAL(75) STACK=0000H				
77	0000H	OPENSYMBOL.	PROCEDURE EXTERNAL(79) STACK=0000H				
8		OPERANDSTRUC.	LITERALLY				
3		OPERATOR.	LITERALLY				
7		OPTR.	LITERALLY				
7		OSEG.	LITERALLY				
7		OSHL.	LITERALLY				
7		OSHORT.	LITERALLY				
7		OSHR.	LITERALLY				
7		OTYPE.	LITERALLY				
10	0000H	1 OUT.	BYTE EXTERNAL(51)	153			
155	0000H	36 OUTBYT.	PROCEDURE STACK=0008H		292	293	309 323
			336 343				
320	043DH	32 OUTBYTES.	PROCEDURE STACK=0010H		335	337	342 345
			347 349				
291	034CH	17 OUTCRLF.	PROCEDURE STACK=000CH		303	333	350 357
			365 367				
56	0000H	OUTHEXBYTE.	PROCEDURE EXTERNAL(70) STACK=0000H				
59	0000H	OUTPRINTBYTE.	PROCEDURE EXTERNAL(71) STACK=0000H				
62	0000H	OUTSYMBOLBYTL.	PROCEDURE EXTERNAL(72) STACK=0000H			165	314
			316 363				
20	0000H	OUTTEXT.	PROCEDURE EXTERNAL(58) STACK=0000H			196	
153	0084H	2 OVERFLOWLTHIT.	WORD 194 239 379				
7		OXOR.	LITERALLY				
327	0004H	2 P.	WORD PARAMETER AUTOMATIC		328	329	330 332
			334 337				
320	0004H	2 P.	WORD PARAMETER AUTOMATIC		321	323	324
96	0000H	1 PAGETEXT.	BYTE ARRAY(1) EXTERNAL(91)				
96	0000H	1 PARERRTEXT.	BYTE ARRAY(1) EXTERNAL(95)				
96	0000H	1 PASSOTEXT.	BYTE ARRAY(1) EXTERNAL(103)				
96	0000H	1 PASSITEXT.	BYTE ARRAY(1) EXTERNAL(104)				
6		PCODEMACRO.	LITERALLY				
6		PCSEG.	LITERALLY				
6		PDB.	LITERALLY				
6		PDBIT.	LITERALLY				
6		PDD.	LITERALLY				
6		PDSEG.	LITERALLY				
6		PDW.	LITERALLY				
6		PEJECT.	LITERALLY				
6		PEND.	LITERALLY				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

	PENDIF	LITERALLY			
	PENOM	LITERALLY			
	PERU	LITERALLY			
	PESEG	LITERALLY			
	PIF	LITERALLY			
	PTFLIST	LITERALLY			
	PINCLUDE	LITERALLY			
	PLIST	LITERALLY			
	PMDRM	LITERALLY			
	PNOIFLIST	LITERALLY			
	PNOLIST	LITERALLY			
	PNOSEGFIX	LITERALLY			
10	0000H 1 POP	BYTE EXTERNAL(43)	153		
10	0000H 1 POWP	BYTE EXTERNAL(49)	153		
	POAG	LITERALLY			
	PPAGESIZE	LITERALLY			
	PPAGENWIDTH	LITERALLY			
	PRB	LITERALLY			
	PRELB	LITERALLY			
	PRELW	LITERALLY			
	PRINTER	LITERALLY			
327	045DH 96 PRINTSYM	PROCEDURE STACK=0016H	353		
295	035DH 124 PRINTSYMBOLS	PROCEDURE STACK=001AH	364	366	368
2	PROCC	LITERALLY	11	14	35
			56	59	62
			65	67	69
			79	81	83
			85	87	89
			91	94	155
			167	176	192
			208	212	221
			225	229	233
			237	245	291
			295	297	320
			327	359	370
	PRS	LITERALLY			
	PRW	LITERALLY			
	PSEGFIX	LITERALLY			
	PSEUDO	LITERALLY			
	PSINFORM	LITERALLY			
	PSSEG	LITERALLY			
167	0004H 2 PT	WORD PARAMETER AUTOMATIC	168	172	
26	0000H 2 PT	WORD PARAMETER AUTOMATIC	27		
176	000AH 2 PT1	WORD PARAMETER AUTOMATIC	177	178	182
192	000AH 2 PT1	WORD PARAMETER AUTOMATIC	193	199	204
192	0008H 2 PT2	WORD PARAMETER AUTOMATIC	193	199	
176	0008H 2 PT2	WORD PARAMETER AUTOMATIC	177	184	
192	0004H 2 PT3	WORD PARAMETER AUTOMATIC	193	202	
176	0006H 2 PT3	WORD PARAMETER AUTOMATIC	177	178	
	PTITLE	LITERALLY			
237	0004H 2 PTR	WORD PARAMETER AUTOMATIC	238	241	
10	0000H 1 PUSH	BYTE EXTERNAL(O)	153		
10	0000H 1 PUSHF	BYTE EXTERNAL(4)	153		
5	RAN	LITERALLY			
5	RAL	LITERALLY			
5	RAX	LITERALLY			
5	RBH	LITERALLY			
5	RBL	LITERALLY			
5	RBP	LITERALLY			
5	RBX	LITERALLY			
5	RCH	LITERALLY			
5	RCL	LITERALLY			
5	RCS	LITERALLY			
5	RCX	LITERALLY			

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

5		RDE.	LITERALLY	
5		RDI.	LITERALLY	
5		RDL.	LITERALLY	
5		RDS.	LITERALLY	
5		RDX.	LITERALLY	
109	0000H	READCONSOLE.	PROCEDURE BYTE EXTERNAL(112) STACK=0000H	
136	0000H	READRECORD.	PROCEDURE BYTE EXTERNAL(122) STACK=0000H	
2		REENT.	LITERALLY	
3		REG.	LITERALLY	
10	0000H	1 REPNZ.	BYTE EXTERNAL(12)	153
10	0000H	1 REPZ.	BYTE EXTERNAL(1)	153
5		RES.	LITERALLY	
233	0004H	2 RESULT.	WORD PARAMETER AUTOMATIC	234 235
221	0004H	2 RESULT.	WORD PARAMETER AUTOMATIC	222 223
10	0000H	1 RET.	BYTE EXTERNAL(39)	153.
213	0090H	2 RETPT.	WORD 213 214 216	
208	0004H	2 RETURNPT.	WORD PARAMETER AUTOMATIC	209 210
245	0004H	2 RETURNPTR.	WORD PARAMETER AUTOMATIC	246 247
81	0000H	REWINDSOURCE.	PROCEDURE EXTERNAL(81) STACK=0000H	
7		RIGHTBRACKET.	LITERALLY	
5		RSI.	LITERALLY	
5		RSP.	LITERALLY	
5		RSS.	LITERALLY	
44	0000H	2 S.	WORD PARAMETER	45
17	0000H	2 S.	WORD PARAMETER	18
10	0000H	1 SAR.	BYTE EXTERNAL(34)	153
176	0056H	97 SEARCH.	PROCEDURE BYTE STACK=0014H	223 235
8		SEGBIT.	LITERALLY	
8		SEGMCOUNT.	LITERALLY	
230	0004H	2 SEGMENT.	WORD MEMBER(SYMB)	
226	0004H	2 SEGMENT.	WORD MEMBER(SYMB)	
193	0007H	2 SEGMENT.	WORD MEMBER(SYMB)	
177	0007H	2 SEGMENT.	WORD MEMBER(SYMBHEAD)	
154	0007H	2 SEGMENT.	WORD MEMBER(SORTSYM)	
154	0007H	2 SEGMENT.	WORD MEMBER(ALPHASYM)	
372	0007H	2 SEGMENT.	WORD MEMBER(SYMB)	
328	0007H	2 SEGMENT.	WORD MEMBER(X)	
8		SEGTYPBIT.	LITERALLY	
8		SEGTYPESCOUNT.	LITERALLY	
121	0000H	SELECTDISK.	PROCEDURE EXTERNAL(117) STACK=0000H	
147	0000H	SETDMAADDRESS.	PROCEDURE EXTERNAL(126) STACK=0000H	
230	0003H	1 SFLAG.	BYTE MEMBER(SYMB)	
226	0003H	1 SFLAG.	BYTE MEMBER(SYMB)	
193	0006H	1 SFLAG.	BYTE MEMBER(SYMB)	
177	0006H	1 SFLAG.	BYTE MEMBER(SYMBHEAD)	
154	0006H	1 SFLAG.	BYTE MEMBER(SORTSYM)	
154	0006H	1 SFLAG.	BYTE MEMBER(ALPHASYM)	
372	0006H	1 SFLAG.	BYTE MEMBER(SYMB)	
328	0006H	1 SFLAG.	BYTE MEMBER(X)	
		SIZE.	BUILTIN 375	
193	0002H	1 SLENGTH.	BYTE MEMBER(SYMB)	203
177	0002H	1 SLENGTH.	BYTE MEMBER(SYMBHEAD)	180
154	0002H	1 SLENGTH.	BYTE MEMBER(SORTSYM)	255
154	0002H	1 SLENGTH.	BYTE MEMBER(ALPHASYM)	256
372	0002H	1 SLENGTH.	BYTE MEMBER(CODM)	
372	0002H	1 SLENGTH.	BYTE MEMBER(SYMB)	
328	0002H	1 SLENGTH.	BYTE MEMBER(X)	332 337

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

252	0000H	1	SORTNAME	BYTE BASED(CS:PTRNAMEPTR) ARRAY(1)	253	260
252	0096H	2	SORTNAMEPTR	WORD 252 253 254 260		
154	0000H	12	SORTSYM	STRUCTURE BASED(SORTSYMPTR)	255	266 267
				285		
249	024CH	68	SORTSYMBOLS	PROCEDURE STACK=000AH	362	
154	008AH	2	SORTSYMPTR	WORD 154 253 255 266 267	283	284 285
				287		
229	0004H	2	SOURCE	WORD PARAMETER AUTOMATIC	230	231
2			SPACE	LITERALLY 309 336 343		
3			SPEC	LITERALLY		
10	0000H	1	STI	BYTE EXTERNAL(44)	153	
233	0006H	2	STRADR	WORD PARAMETER AUTOMATIC	234	235
221	0006H	2	STRADR	WORD PARAMETER AUTOMATIC	222	223
3			STRING	LITERALLY		
2			STRUC	LITERALLY 154 177 193 226 230	328	372
230	0002H	1	STYPE	BYTE MEMBERCSYMB)		
226	0002H	1	STYPE	BYTE MEMBERCSYMB)		
193	0005H	1	STYPE	BYTE MEMBERCSYMB)		
177	0005H	1	STYPE	BYTE MEMBERCSYMBHEAD)		
154	0005H	1	STYPE	BYTE MEMBERCSORTSYM)		
154	0005H	1	STYPE	BYTE MEMBER(ALPHASYM)		
372	0005H	1	STYPE	BYTE MEMBERCSYMB)		
328	0005H	1	STYPE	BYTE MEMBER(X)	330	
1	0000H		SYMB	PROCEDURE STACK=0000H		
230	0000H	9	SYMB	STRUCTURE BASED(SYMBADR)	231	
226	0000H	9	SYMB	STRUCTURE BASED(SYMBADR)	227	
193	0000H	12	SYMB	STRUCTURE BASED(FREET)	200	203
372	00ADH	12	SYMB	STRUCTURE AT 376		
229	0006H	2	SYMBADR	WORD PARAMETER AUTOMATIC	230	231
225	0006H	2	SYMBADR	WORD PARAMETER AUTOMATIC	226	227
153	0000H	128	SYMBENTRY	WORD ARRAY(64)	210 223	283 375
96	0000H	1	SYMBFULLERRTEXT	BYTE ARRAY(1) EXTERNAL(101)		196
177	0000H	12	SYMBHEAD	STRUCTURE BASED(CURRENTPT)		180
370	04ECH	74	SYMBINIT	PROCEDURE PUBLIC STACK=000AH		
3			SYMBOL	LITERALLY		
152	0000H	1	SYMBOLDEVICE	BYTE EXTERNAL(128)	306	360
8			SYMBOLHEAD	LITERALLY 154 177 193 328	372	
153	009EH	1	SYMBOLHEADLG	BYTE 210 223 253 254 329	376	377 379
8			SYMBOLSTRUC	LITERALLY 226 230		
177	0000H	2	SYMBPTR	WORD BASED(PT?)	184	
153	0000H	1	SYMBTAB	BYTE ARRAY(1) AT	374	
359	04BDH	47	SYMBTERMINATE	PROCEDURE PUBLIC STACK=001EH		
193	0000H	2	SYMPTR	WORD BASED(PT3)	202	
107	0000H		SYSTEMRESET	PROCEDURE EXTERNAL(111) STACK=0000H		197
20	0000H	2	T	WORD PARAMETER	21	
2			TAB	LITERALLY 314 316		
105	0000H	1	TBUFF	BYTE ARRAY(1) EXTERNAL(109)		
23	0000H	2	TEXTATK	WORD PARAMETER	24	
2			THLNDG	LITERALLY 157 180 182 194 214	272	301
				306 330		
2			TRUE	LITERALLY 185 206 217 243 261	263	
295	009AH	1	TYP	BYTE PARAMETER	295 330 344 346	348
8			TYPEBTI	LITERALLY		
11	0000H		TYPECALC	PROCEDURE BYTE EXTERNAL(55) STACK=0000H		
8			TYPECOUNT	LITERALLY		
3			UDFSYMB	LITERALLY		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

41	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(65) STACK=0000H
96	0000H	1	USREAKEXT.	BYTE ARRAY(1) EXTERNAL(102)
96	0000H	1	USLFACTOR.	BYTE ARRAY(1) EXTERNAL(93)
11	0000H	2	VAL.	WORD PARAMETER 12
9			VALIDDISK.	LITERALLY
3			VARIABLE.	LITERALLY 346 364
119	0000H		VERSION.	PROCEDURE WORD EXTERNAL(116) STACK=0000H
4			WRD.	LITERALLY
14	0000H		WROTEST.	PROCEDURE BYTE EXTERNAL(56) STACK=0000H
111	0000H		WRITECONSOLE.	PROCEDURE EXTERNAL(113) STACK=0000H
114	0000H		WRITELIST.	PROCEDURE EXTERNAL(114) STACK=0000H
139	0000H		WRITERECORD.	PROCEDURE BYTE EXTERNAL(123) STACK=0000H
328	0000H	12	X.	STRUCTURE BASED(P) 330 332 334 337
298	00A7H	1	X.	BYTE 299 301
10	0000H	1	XCHG.	BYTE EXTERNAL(38) 153
10	0000H	1	XLAT.	BYTE EXTERNAL(52) 153
298	00A8H	1	Y.	BYTE 300 308 313 315
96	0000H	1	YES.	BYTE EXTERNAL(88)

MODULE INFORMATION:

CODE AREA SIZE = 0536H 13340
 CONSTANT AREA SIZE = 009AH 1540
 VARIABLE AREA SIZE = 00AEH 1740
 MAXIMUM STACK SIZE = 001EH 300
 946 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-II PL/M-86 V2.0 COMPILETIME OF MODULE TO
OBJECT MODULE PLACED IN IO.OBJ

COMPILER INVOKED BY: :F0: IO.PLN DEBUG OPTIMIZE(2) DATE(10/5/81) PAGELWIDTH(100) XREF

```

1      ;title ("INTERFACE TO CP/M I/O")
      io:
      oo:

      /*
      Template for all BIOS calls
      */

      /*

      modified 3/26/81 R. Silberstein
      modified 6/16/81 R. Silberstein
      modified 9/14/81 R. Silberstein

      */

2      1      declare tbuff (80h) byte external;

3      1      mon1: procedure (func,info) external;
4      2      declare func byte;
5      2      declare info address;
6      2      end mon1;

7      1      mon2: procedure (func,info) byte external;
8      2      declare func byte;
9      2      declare info address;
10     2      end mon2;

      /*****
      *
      *      B I O S      Externals
      *
      *****/

11     1      system$reset: procedure public;
12     2      call mon1 (0,0);
13     2      end system$reset;

14     1      read$console: procedure byte public;
15     2      return mon2 (1,0);
16     2      end read$console;

17     1      write$console: procedure (char) public;
18     2      declare char byte;
19     2      call mon1 (2,char);
20     2      end write$console;

21     1      write$list: procedure (char) public;
22     2      declare char byte;
23     2      call mon1 (5,char);
24     2      end write$list;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
25 1  constat: procedure byte public;
26 2    return mon2 (11,0);
27 2  end constat;

28 1  VERSION: PROCEDURE ADDRESS PUBLIC;
29 2    RETURN MON2 (12, 0);
30 2  END VERSION;

31 1  select$disk: procedure (disk$number) public;
32 2    declare disk$number byte;
33 2    call mon1 (14,disk$number);
34 2  end select$disk;

35 1  set$DMA$address: procedure (DMA$address) public;
36 2    declare DMA$address address;
37 2    call mon1 (26,DMA$address);
38 2  end set$DMA$address;

39 1  open$file: procedure (fcb$address) byte public;
40 2    declare fcb$address address;
41 2    CALL SET$DMA$ADDRESS (.TBUFF); /* FOR 1.4 SYSTEMS */
42 2    return mon2 (15,fcb$address);
43 2  end open$file;

44 1  OPEN$RO$FILE: PROCEDURE (FCB$ADDRESS) BYTE PUBLIC;
45 2    DECLARE FCB$ADDRESS ADDRESS, FCB BASED FCB$ADDRESS (32) BYTE;
46 2    FCB (6) = FCB (6) OR 80H;
47 2    RETURN OPEN$FILE (FCB$ADDRESS);
48 2  END OPEN$RO$FILE;

49 1  close$file: procedure (fcb$address) byte public;
50 2    declare fcb$address address;
51 2    return mon2 (16,fcb$address);
52 2  end close$file;

53 1  delete$file: procedure (fcb$address) public;
54 2    declare fcb$address address;
55 2    CALL mon1 (19,fcb$address);
56 2  end delete$file;

57 1  read$record: procedure (fcb$address) byte public;
58 2    declare fcb$address address;
59 2    return mon2 (20,fcb$address);
60 2  end read$record;

61 1  write$record: procedure (fcb$address) byte public;
62 2    declare fcb$address address;
63 2    return mon2 (21,fcb$address);
64 2  end write$record;

65 1  create$file: procedure (fcb$address) byte public;
66 2    declare fcb$address address;
67 2    return mon2 (22,fcb$address);
68 2  end create$file;

69 1  interrogate$disk: procedure byte public;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
70 2      return mon2 (25,0);
71 2      end interrogate$disk;

72 1      crlf: procedure public;
73 2          call write$console (0dh);
74 2          call write$console (0ah);
75 2      end crlf;

76 1      end io;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES		
21	0004H	1	CHAR	BYTE PARAMETER AUTOMATIC	22 23
17	0004H	1	CHAR	BYTE PARAMETER AUTOMATIC	18 19
49	0001H	16	CLOSEFILE	PROCEDURE BYTE PUBLIC STACK=000AH	
25	0043H	15	CONSTAT.	PROCEDURE BYTE PUBLIC STACK=0008H	
65	00F1H	16	CREATEFILE	PROCEDURE BYTE PUBLIC STACK=000AH	
72	0110H	17	CRLF	PROCEDURE PUBLIC STACK=000EH	
53	00C1H	16	DELETEFILE	PROCEDURE PUBLIC STACK=000AH	
31	0004H	1	DISKNUMBER	BYTE PARAMETER AUTOMATIC	32 33
35	0004H	2	DMAADDRESS	WORD PARAMETER AUTOMATIC	36 37
45	0000H	32	FCB	BYTE BASED(FCBADDRESS) ARRAY(32)	46
61	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	62 63
44	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	45 46 47
39	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	40 42
49	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	50 51
53	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	54 55
65	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	66 67
57	0004H	2	FCBADDRESS	WORD PARAMETER AUTOMATIC	58 59
3	0000H	1	FUNC	BYTE PARAMETER	4
7	0000H	1	FUNC	BYTE PARAMETER	8
7	0000H	2	INFO	WORD PARAMETER	9
3	0000H	2	INFO	WORD PARAMETER	5
69	0101H	15	INTERROGATEDISK . .	PROCEDURE BYTE PUBLIC STACK=0008H	
1	0000H		ID	PROCEDURE STACK=0000H	
3	0000H		MON1	PROCEDURE EXTERNAL(1) STACK=0000H	12 19
				23 33 37 55	
7	0000H		MON2	PROCEDURE BYTE EXTERNAL(2) STACK=0000H	15
				26 29 42 51 59 63 67 70	
39	0086H	23	OPENFILE	PROCEDURE BYTE PUBLIC STACK=0010H	47
44	009DH	20	OPENKOFLE	PROCEDURE BYTE PUBLIC STACK=0016H	
14	000EH	15	READCONSOLE	PROCEDURE BYTE PUBLIC STACK=0008H	
57	00D1H	16	READRECORD	PROCEDURE BYTE PUBLIC STACK=000AH	
31	0063H	19	SELECTDISK	PROCEDURE PUBLIC STACK=000AH	
35	0076H	16	SETDMAADDRESS	PROCEDURE PUBLIC STACK=000AH	41
11	0000H	14	SYSTEMRESET	PROCEDURE PUBLIC STACK=0008H	
2	0000H	128	TBUFF.	BYTE ARRAY(128) EXTERNAL(0)	41
28	0052H	17	VERSION.	PROCEDURE WORD PUBLIC STACK=0008H	
17	001DH	19	WRITECONSOLE	PROCEDURE PUBLIC STACK=000AH	73 74
21	0030H	19	WRITELIST.	PROCEDURE PUBLIC STACK=000AH	
61	00E1H	16	WRITERECORD.	PROCEDURE BYTE PUBLIC STACK=000AH	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

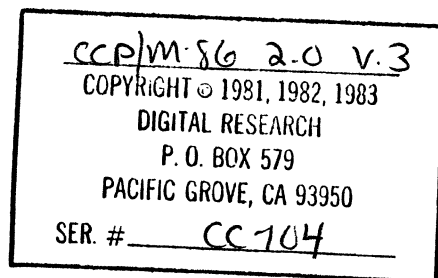
SER. # _____

MODULE INFORMATION:

CODE AREA SIZE = 0121H 2890
 CONSTANT AREA SIZE = 0000H 00
 VARIABLE AREA SIZE = 0000H 00
 MAXIMUM STACK SIZE = 0016H 220
 117 LINES READ

0 PROGRAM ERRORS)

END OF PL/M-86 COMPILATION





ASIS-II PL/M-86 V2.0 COMPTLATION OF MODULE SUBR1

OBJECT MODULE PLACED IN SUBR1.OBJ

COMPILER INVOKED BY: %F0: SUBR1.PL4 DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

```

1      $title ("SUBROUTINE MODULE - PART 1")
      subr1:
      do;

      $include (:f1:macro.lit)
=     $nolist

      /*

      modified 3/26/81 R. Silberstein

      */

      /* compute if number is in range (-128,127) */
      /* exit 1 if in range, 2 otherwise */

3      1      typecalc: procedure(val) byte public;
4      2      declare val address,
              lowb byte at (.val),
              highb byte at (.val+1);
5      2      lowb=lowb and 80h;
6      2      if highb=0 then
7      2          if lowb=0 then return 1;
9      2      if highb=0ffh then
10     2          if lowb <> 0 then return 1;
12     2      return 2;
13     2      end typecalc;

      /* test if number is a "word" (>255 and <-256) */
14     1      wrdtest: procedure(n) byte public;
15     2      declare n address;
16     2      return ((n < 0ff00h) and (n > 0ffh));
17     2      end wrdtest;

18     1      copy: procedure(n,s,d) public;
19     2      declare n byte,
              (s,d) address,
              sch based s byte,
              dch based d byte;
20     2      DO WHILE (N := N - 1) <> 0FFH;
21     3          DCH = SCH;
22     3          D = D + 1;
23     3          S = S + 1;
24     3      END;
25     2      end copy;

26     1      end subr1;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
------	------	------	----------------------------------

2			ADDR LITERALLY
18	0048H	37	COPY PROCEDURE PUBLIC STACK=0008H
2			CR LITERALLY
18	0004H	2	D. WORD PARAMETER AUTOMATIC 19 21 22
19	0000H	1	DCH. BYTE BASED(C) 21
2			DCL. LITERALLY
2			ELSEDD LITERALLY
2			ENDCASE. LITERALLY
2			ENDDO. LITERALLY
2			ENDFOREVER LITERALLY
2			ENDIF. LITERALLY
2			ENDMODULE. LITERALLY
2			ENDOFFILE. LITERALLY
2			ENDPROC. LITERALLY
2			ENDWHILE LITERALLY
2			FALSE. LITERALLY
2			FOREVER. LITERALLY
2			FORMFEED LITERALLY
4	0005H	1	HIGHB. BYTE AT AUTOMATIC 6 9
2			INIT LITERALLY
2			LF LITERALLY
2			LIT. LITERALLY 2
4	0004H	1	LOWB. BYTE AT AUTOMATIC 5 7 10
14	0004H	2	N. WORD PARAMETER AUTOMATIC 15 16
18	0008H	1	N. BYTE PARAMETER AUTOMATIC 19 20
2			PROC LITERALLY
2			REENT. LITERALLY
18	0006H	2	S. WORD PARAMETER AUTOMATIC 19 21 23
19	0000H	1	SCH. BYTE BASED(S) 21
2			SPACE. LITERALLY
2			STRUC. LITERALLY
1	0000H		SUBRI. PROCEDURE STACK=0000H
2			TAB. LITERALLY
2			THENDO LITERALLY
2			TRUE LITERALLY
3	0000H	41	TYPECALC PROCEDURE BYTE PUBLIC STACK=0004H
3	0004H	2	VAL. WORD PARAMETER AUTOMATIC 4
14	0029H	31	WRDTEST. PROCEDURE BYTE PUBLIC STACK=0006H

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

MODULE INFORMATION:

CODE AREA SIZE = 006DH 109D
 CONSTANT AREA SIZE = 0000H 0D
 VARIABLE AREA SIZE = 0000H 0D
 MAXIMUM STACK SIZE = 0008H 8D
 80 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE SUBR2

OBJECT MODULE PLACED IN SUBR2.OBJ

COMPILER INVOKED BY: :F0: SUBR2.PLM DEBJG OPTIMIZE(2) DATE(10/5/81) PAGESIZE(100) XREF

```

1      $title ("SUBROUTINE MODULE - PART 2")
      subr2:
      do:
      $include(:f1:macro.lit)
=     $nolist
      $INCLUDE (:F1:STRUC.LIT)
=     $nolist
      $include(:f1:io.ext)
=     $nolist

/*

      modified 3/26/81 R. Silberstein
      modified 3/28/81 R. Silberstein
      modified 3/30/81 R. Silberstein

*/

59 1      outtext: procedure (t) public;
60 2      dcl t addr,
        ch based t byte;
61 2      do while ch <> 0;
62 3          call write$console(ch);
63 3          t=t+1;
64 3      end$while;
65 2      end outtext;

66 1      OUTFILENAME: PROCEDURE (PTR);
67 2      DECLARE PTR ADDRESS, X BASED PTR FILEDSTRUCTURE, I BYTE;
68 2      CALL WRITE$CONSOLE (X.DISK + "A");
69 2      CALL WRITE$CONSOLE (" ");
70 2      DO I = 1 TO 8;
71 3          IF (X.FCBBLOCK (I) AND 7FH) = SPACE THEN I = 8;
72 3          ELSE CALL WRITE$CONSOLE (X.FCBBLOCK (I) AND 7FH);
73 3      END;
74 2      CALL WRITE$CONSOLE (" ");
75 2      DO I = 9 TO 11;
76 3          CALL WRITE$CONSOLE (X.FCBBLOCK (I) AND 7FH);
77 3      END;
78 2      CALL SYSTEMRESET;
79 2      END OUTFILENAME;
80 2

81 1      FILEABORT: PROCEDURE (PTR, TEXTADR) PUBLIC;
82 2      DECLARE (PTR, TEXTADR) ADDRESS;
83 2      CALL OUTTEXT (TEXTADR);
84 2      CALL WRITE$CONSOLE (" ");
85 2      CALL WRITE$CONSOLE (SPACE);
86 2      CALL OUTFILENAME (PTR);
87 2      END FILEABORT;

88 1      fill: procedure (ch,n,pt) public;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
89 2      dcl (ch,n) byte,pt address,buffer based pt byte;
90 2      DO WHILE (N := N - 1) <> OFFn;
91 3          buffer=ch;
92 3          pt = pt + 1;
93 3      end$while;
94 2      end fill;

95 1      digit: procedure(ch) byte public;
96 2          dcl ch byte;
97 2          IF CH < "0" THEN RETURN FALSE;
99 2          return (ch <= "9");
100 2      end digit;

101 1      letter: procedure(ch) byte public;
102 2          dcl ch byte;
103 2          IF CH < "A" THEN RETURN FALSE;
105 2          return (ch <= "Z");
106 2      end letter;

107 1      alphanumeric: proc(ch) byte public;
108 2          dcl ch byte;
109 2          if letter(ch) then return true;
111 2          return digit(ch);
112 2      end alphanumeric;

113 1      asciichar: proc (ch) byte public;
114 2          dcl ch byte;
115 2          if ch=cr then return true;
117 2          IF CH = LF THEN RETURN TRUE;
119 2          IF CH < SPACE THEN RETURN FALSE;
121 2          return (ch <= 7eh);
122 2      end asciichar;

123 1      upper: procedure(ch) byte public;
124 2          dcl ch byte;
125 2          if ch >= 61h THEN IF ch <= 7eh then ch=cn-20h;
128 2          return ch;
129 2      end upper;

130 1      equal: procedure(n,s,d) byte public;
131 2          dcl n byte,
              (s,d) address,
              sch based s byte,
              dch based d byte;
132 2          DO WHILE (N := N - 1) <> OFFH;
133 3              IF SCH <> DCH THEN RETURN FALSE;
135 3              S = S + 1;
136 3              D = D + 1;
137 3          END$WHILE;
138 2          return true;
139 2      end equal;

140 1      hexlout: procedure(n,d) public;
141 2          dcl n byte,d addr,
              dest based d (1) byte;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

142 2      hexdigit: procedure(digit) byte;
143 3          dcl digit byte;
144 3          digit=digit+"0";
145 3          if digit > "9" then digit=digit+7;
147 3          return digit;
148 3      end hexdigit;

149 2      dest(0)=hexdigit(SHR (N, 4));
150 2      dest(1)=hexdigit(n and 0fh);
151 2      end hexlout;

152 1      hex2out: proc (n,d) public;
153 2          dcl n addr,
154 2              d addr;
154 2          call hexlout(HIGH (N),d);
155 2          call hexlout(LOW (N),d+2);
156 2      end hex2out;

157 1      decout: proc (n,d) public;
158 2          dcl
159 2              n      addr,
160 2              d      address,
161 2              dest based d (1) byte,
162 2              (i,space$or$zero,digit) byte,
163 2              divis(5) addr data (10000,1000,100,10,1);
159 2          space$or$zero=space;
160 2          do i=0 to 4;
161 3              if i=4 then space$or$zero="0";
163 3              digit=n/divis(i);
164 3              n=n mod divis(i);
165 3              if digit=0 then$do
167 4                  dest(i)=space$or$zero;
168 4              else$do
170 4                  dest(i)=digit+"0";
171 4                  space$or$zero="0";
172 4              end$if;
173 3          end$do;
174 2      end decout;

175 1      end subr2;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
2			ADDR LITERALLY 60 67 141 153 158
107	0111H	27	ALPHANUMERIC PROCEDURE BYTE PUBLIC STACK=0004H
113	012CH	42	ASCIICHAR PROCEDURE BYTE PUBLIC STACK=0004H
3			BASEREGBIT LITERALLY
3			BASEREGCOUNT LITERALLY
3			BREGBIT LITERALLY
3			BREGCOUNT LITERALLY
89	0000H	1	BUFFER BYTE BASED(PT) 91
67	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(X)
67	0022H	2	BUFPTR WORD MEMBER(X)
123	0004H	1	CH BYTE PARAMETER AUTOMATIC 124 125 126 127
			128
113	0004H	1	CH BYTE PARAMETER AUTOMATIC 114 115 117 119
			121
60	0000H	1	CH BYTE BASED(CT) 61 62
88	0008H	1	CH BYTE PARAMETER AUTOMATIC 89 91
95	0004H	1	CH BYTE PARAMETER AUTOMATIC 96 97 99
107	0004H	1	CH BYTE PARAMETER AUTOMATIC 108 109 111
101	0004H	1	CH BYTE PARAMETER AUTOMATIC 102 103 105
21	0000H	1	CHAR BYTE PARAMETER 22
18	0000H	1	CHAR BYTE PARAMETER 19
37	0000H		CLOSEFILE PROCEDURE BYTE EXTERNAL(15) STACK=0000H
3			CODEMACROHEAD LITERALLY
24	0000H		CONSTAT PROCEDURE BYTE EXTERNAL(10) STACK=0000H
2			CR LITERALLY 115
49	0000H		CREATEFILE PROCEDURE BYTE EXTERNAL(19) STACK=0000H
57	0000H		CRLF PROCEDURE EXTERNAL(22) STACK=0000H
140	0004H	2	D WORD PARAMETER AUTOMATIC 141 149 150
152	0004H	2	D WORD PARAMETER AUTOMATIC 153 154 155
130	0004H	2	D WORD PARAMETER AUTOMATIC 131 133 136
157	0004H	2	D WORD PARAMETER AUTOMATIC 158 157 170
131	0000H	1	DCH BYTE BASED(D) 133
2			DCL LITERALLY
157	01FEH	122	DECOUT PROCEDURE PUBLIC STACK=0006H
40	0000H		DELETEFILE PROCEDURE EXTERNAL(16) STACK=0000H
158	0000H	1	DEST BYTE BASED(D) ARRAY(1) 167 170
141	0000H	1	DEST BYTE BASED(D) ARRAY(1) 149 150
158	0000H	1	DIGIT BYTE 163 165 170
95	0000H	26	DIGIT PROCEDURE BYTE PUBLIC STACK=0004H 111
142	0004H	1	DIGIT BYTE PARAMETER AUTOMATIC 143 144 145 146
			147
67	0000H	1	DISK BYTE MEMBER(X) 68
28	0000H	1	DISKNUMBER BYTE PARAMETER 29
158	0000H	10	DIVTS WORD ARRAY(5) DATA 163 164
54	0000H	2	DMAADDRESS WORD PARAMETER 55
2			ELSEDO LITERALLY
13	0000H	2	ENDBUF WORD EXTERNAL(5)
2			ENDCASE LITERALLY
2			ENDDU LITERALLY

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

2		ENDFOREVER	LITERALLY		
2		ENDIF	LITERALLY		
2		ENDMODULE	LITERALLY		
2		ENDOFFILE	LITERALLY		
2		ENDPROC	LITERALLY		
2		ENDWHILE	LITERALLY		
130	0170H	45 EQUAL	PROCEDURE BYTE PUBLIC STACK=0008H		
2		FALSE	LITERALLY	98 104 120 134	
10	0000H	1 FCB	BYTE ARRAY(1) EXTERNAL(2)		
11	0000H	1 FCB16	BYTE ARRAY(1) EXTERNAL(3)		
49	0000H	2 FCBADDRESS	WORD PARAMETER	50	
31	0000H	2 FCBADDRESS	WORD PARAMETER	32	
43	0000H	2 FCBADDRESS	WORD PARAMETER	44	
40	0000H	2 FCBADDRESS	WORD PARAMETER	41	
37	0000H	2 FCBADDRESS	WORD PARAMETER	38	
34	0000H	2 FCBADDRESS	WORD PARAMETER	35	
46	0000H	2 FCBADDRESS	WORD PARAMETER	47	
67	0001H	33 FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	71 73 77	
81	009EH	31 FILEABORT	PROCEDURE PUBLIC STACK=0010H		
3		FILEISTRUCTURE	LITERALLY		
3		FILEOSTRUCTURE	LITERALLY	67	
88	00BDH	32 FILL	PROCEDURE PUBLIC STACK=0008H		
2		FOREVER	LITERALLY		
2		FORMFEED	LITERALLY		
7	0000H	1 FUNC	BYTE PARAMETER	9	
4	0000H	1 FUNC	BYTE PARAMETER	5	
140	0190H	38 HEX1OUT	PROCEDURE PUBLIC STACK=000CH	154 155	
152	01DDH	33 HEX2OUT	PROCEDURE PUBLIC STACK=0014H		
142	01C3H	26 HEXDIGIT	PROCEDURE BYTE STACK=0004H	149 150	
		HIGH	BUILTIN	154	
158	0001H	1 I	BYTE	160 161 163 164 167 170	
67	0000H	1 I	BYTE	70 71 72 73 76 77	
3		INDEXREGBIT	LITERALLY		
3		INDEXREGCOUNT	LITERALLY		
7	0000H	2 INFO	WORD PARAMETER	8	
4	0000H	2 INFO	WORD PARAMETER	5	
2		INIT	LITERALLY		
52	0000H	INTERROGATEDISK	PROCEDURE BYTE EXTERNAL(20) STACK=0000H		
3		IREGBIT	LITERALLY		
3		IREGCOUNT	LITERALLY		
101	00F7H	26 LETTER	PROCEDURE BYTE PUBLIC STACK=0004H	109	
2		LF	LITERALLY	117	
2		LIT	LITERALLY	2 3	
		LOW	BUILTIN	155	
4	0000H	MON1	PROCEDURE EXTERNAL(0) STACK=0000H		
7	0000H	MON2	PROCEDURE BYTE EXTERNAL(1) STACK=0000H		
157	0006H	2 N	WORD PARAMETER AUTOMATIC	158 163 164	
152	0006H	2 N	WORD PARAMETER AUTOMATIC	153 154 155	
140	0006H	1 N	BYTE PARAMETER AUTOMATIC	141 149 150	
130	0008H	1 N	BYTE PARAMETER AUTOMATIC	131 132	
88	0006H	1 N	BYTE PARAMETER AUTOMATIC	89 90	
3		NOOVERRCOUNT	LITERALLY		
3		NOOVERRIDEBIT	LITERALLY		
31	0000H	OPENFILE	PROCEDURE BYTE EXTERNAL(13) STACK=0000H		
34	0000H	OPENROFILE	PROCEDURE BYTE EXTERNAL(14) STACK=0000H		
3		OPERANDSTRUC	LITERALLY		
66	0019H	133 OUTFILENAME	PROCEDURE STACK=0008H	86	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

59	0000H	25	OUTTEXT.	PROCEDURE PUBLIC STACK=0000H	93			
2			PROC.	LITERALLY 107 113 152 157				
88	0004H	2	PT.	WORD PARAMETER AUTOMATIC	89	91	92	
81	0006H	2	PTR.	WORD PARAMETER AUTOMATIC	82	86		
66	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	67	68	71	73
				77				
16	0000H		READCONSOLE. . . .	PROCEDURE BYTE EXTERNAL(7) STACK=0000H				
43	0000H		READRECORD. . . .	PROCEDURE BYTE EXTERNAL(17) STACK=0000H				
2			REENT.	LITERALLY				
130	0006H	2	S.	WORD PARAMETER AUTOMATIC	131	133	135	
131	0000H	1	SCH.	BYTE BASED(S) 133				
3			SEGMBIT.	LITERALLY				
3			SEGMCOUNT.	LITERALLY				
3			SEGTYPEBIT.	LITERALLY				
3			SEGTYPECOUNT. . . .	LITERALLY				
28	0000H		SELECTDISK.	PROCEDURE EXTERNAL(12) STACK=0000H				
54	0000H		SETDMAADDRESS. . . .	PROCEDURE EXTERNAL(21) STACK=0000H				
			SHR.	BUILTIN 149				
2			SPACE.	LITERALLY 71 85 119 159				
158	0002H	1	SPACEORZERO. . . .	BYTE 159 162 167 171				
2			STRUC.	LITERALLY 67				
1	0000H		SUBRZ.	PROCEDURE STACK=0000H				
3			SYMBOLHEAD.	LITERALLY				
3			SYMBOLSTRUC.	LITERALLY				
14	0000H		SYSTEMRESET. . . .	PROCEDURE EXTERNAL(6) STACK=0000H	79			
59	0004H	2	T.	WORD PARAMETER AUTOMATIC	60	61	62	63
2			TAB.	LITERALLY				
12	0000H	1	TBUFF.	BYTE ARRAY(1) EXTERNAL(4)				
81	0004H	2	TEXTADR.	WORD PARAMETER AUTOMATIC	82	83		
2			THEND0.	LITERALLY 165				
2			TRUE.	LITERALLY 110 116 118 138				
3			TYPEBIT.	LITERALLY				
3			TYPECOUNT.	LITERALLY				
123	0156H	26	UPPER.	PROCEDURE BYTE PUBLIC STACK=0004H				
26	0000H		VERSION.	PROCEDURE WORD EXTERNAL(11) STACK=0000H				
18	0000H		WRITECONSOLE. . . .	PROCEDURE EXTERNAL(9) STACK=0000H	62	68		
				69 73 75 77 84 85				
21	0000H		WRITELIST.	PROCEDURE EXTERNAL(9) STACK=0000H				
46	0000H		WRITERECORD. . . .	PROCEDURE BYTE EXTERNAL(18) STACK=0000H				
67	0000H	548	X.	STRUCTURE BASED(PTR) 68 71 73 77				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

MODULE INFORMATION:

CODE AREA SIZE = 0279H 6320
 CONSTANT AREA SIZE = 0004H 100
 VARIABLE AREA SIZE = 0004H 40
 MAXIMUM STACK SIZE = 0014H 200
 372 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

ISIS-II PL/M-86 V2.0 COMPIATION OF MODULE FILE

OBJECT MODULE PLACED IN FILES.DPJ

COMPILER INVOKED BY: :F0: FILES.PLM DLBJG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

\$title("FILE AND I/O MODULE")

file:

do;

/*

modified 3/25/81 R. Silberstein
 modified 3/28/81 R. Silberstein
 modified 3/30/81 R. Silberstein
 modified 4/7/81 R. Silberstein
 modified 4/16/81 R. Silberstein
 modified 6/16/81 R. Silberstein
 modified 9/14/81 R. Silberstein

*/

/*

This is the modules to perform BYTE i/o to
 the following 5 logical devices:

source - file
 include - file
 hex - file
 symbol - file
 print - file

Each of the logical files may be assigned to the
 following physical devices :

null (not legal for source and include file)
 console
 printer (not legal for source and include file)
 disk

The module defines the following set
 of public subroutines:

INSOURCEBYTE	- read 1 byte from source file
ININCLUDEBYTE	- read 1 byte from include file
OUTHEXBYTE (ch)	- write 1 byte to hex file
OUTSYMBOLBYTE (ch)	- write 1 byte to symbol file
OUTPRINTBYTE (ch)	- write 1 byte to print file
OPENSOURCE	- open source file
OPENINCLUDE	- open include file
OPENHEX	- open hex file
OPENSYMBOL	- open symbol file
OPENPRINT	- open print file
REWINDSOURCE	- rewind source file
CLOSESOURCE	- close source file

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

```

CLOSEINCLUDE - close include file
CLOSEHEX     - close hex file
CLOSESYMBOL  - close symbol file
CLOSEPRINT   - close print file

```

In addition, 2 subroutines to set up the correct file names and routing to correct physical device are included. These are:

```

FILESETUP
!FILESETUP

```

The "filesetup" routine sets up the source, hex, symbol and print files by scanning the user command tail of the program activating line. The format of the command line is described in the program format section of the user's manual. The routine also initiates the global string array "SOURCENAME" with the source file name, this array to be used later by the printout module.

The "ifilesetup" sets up the format of the include file given by the INCLUDE command of the assembler.

```

*/
#include (:f1:macro.lit)
= $nolist
#include (:f1:struc.lit)
= $nolist
#include (:f1:dev.lit)
= $nolist
#include (:f1:io.ext)
= $nolist
#include (:f1:subr1.ext)
= $nolist
#include (:f1:subr2.ext)
= $nolist
$INCLUDE (:F1:TEXT.EXT)
= $nolist
#include (:f1:global.ext)
= $nolist

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

109 1  dcl
      diskunit          byte,

      nulltype          lit    '0',    /* subroutine "devicetype" */
      consoletype       lit    '1',
      printertype       lit    '2',
      disktype          lit    '3',

      dr                lit    '0',    /* drive code in fcb block */
      fn                lit    '1',    /* filename in fcb block */
      ft                lit    '9',    /* filetype in fcb block */
      ex                lit    '12',   /* file extension number */
      s2                lit    '14',
      nr                lit    '32',   /* file record number */
      dollar            lit    '---$---',

```

```

asndefault(3)  byte data ('ASB'), /* different file types */
hexdefault(3)  byte data ('HBS'),
lstdefault(3)  byte data ('LST'),
syndefault(3)  byte data ('SYM'),

```

```

sourcefile      file%$structure,
includefile     file%$structure,
hexfile         file%$structure,
printfile       file%$structure,
symbolfile      file%$structure;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

110 1  clearfcb: proc(fcbpt,defaultpt);
111 2      dcl
          (fcbpt,defaultpt) addr,
          dest based fcbpt (1) byte;
112 2      CALL FILL (0, 33, FCBPT);
113 2      CALL FILL (' ', 8, FCBPT+FY);
114 2      call copy(3,defaultpt,.dest(ft));
115 2  end clearfcb;

116 1  clearcontrol: procedure(point,defaultptr);
117 2      dcl (point,defaultptr) addr,
          x based point file%$structure;
118 2      call clearfcb(.x.fcbblock,defaultptr);
119 2      x.disk=diskunit;
120 2  end clearcontrol;

121 1  devicetype: proc(ch) byte;
122 2      dcl ch byte;
123 2      if ch=null then return nulltype;
124 2      if ch=console then return consoletype;
125 2      if ch=printer then return printertype;
126 2      return disktype;
127 2  end devicetype;

131 1  disk$select: procedure(disk);
132 2      dcl disk byte;
133 2      if diskunit <> disk then%do
134 3          diskunit=disk;
135 3          call select$disk(diskunit);
136 3      end%if;
137 2  end disk$select;

139 1  inbyte: proc (ptr) byte;
140 2      dcl ptr addr,
          x based ptr file%$structure,
          ch byte,
          i addr;

141 2      i=x.bufptr;
142 2      if i=length(x.buffer) then%do
143 3          i=0;
144 3          call disk$select(x.disk);
145 3          do while i < length(x.buffer);
146 4              call SET$DMA$ADDRESS (.x.buffer(i));
147 4              IF (CH := READ$RECORD (.X.FCBBLOCK)) <> 0 THEN%DO
148 4

```

```

150 5      IF CH = 1 THEN$DO
152 6          X.BUFFER (1) = END$OF$FILE;
153 6          I = LENGTH (X.BUFFER);
154 6      ELSE$DO
156 6          CALL FILE$ABORT (.X, .DISKREADERRTEXT);
157 6      END$IF;
158 5      ELSE$DO
160 5          I=I+128;
161 5      END$IF;
162 4      END$WHILE;
163 3      I=0;
164 3      END$IF;
165 2      CH=X.BUFFER(I);
166 2      X.BUFPTR=I+1;
167 2      RETURN CH;
168 2      END INBYTE;

169 1      FLUSHBUFFER: PROCEDURE (PTR);
170 2          DECLARE (PTR, I) ADDRESS, X BASED PTR FILE$0$STRUCTURE;

171 2          call disk$select(x.disk);
172 2          I=0;
173 2          DO WHILE I < X.BUFPTR;
174 3              call SET$DMA$ADDRESS (.X.BUFFER(I));
175 3              IF WRITE$RECORD (.X.FCBBLOCK) > 0 THEN
176 3                  CALL FILE$ABORT (.X, .DISKWRITEERRTEXT);
177 3                  I=I+128;
178 3              END$WHILE;
179 2      END FLUSHBUFFER;

180 1      outbyte: proc(ch,ptr);
181 2          dcl ch byte,
              ptr addr,
              x based ptr file$0$structure,
              i addr;

182 2          DO CASE devicetype(x.disk);

183 3              /* null */
              DO; END; /* do nothing */

185 3              /* console */
              call write$console(ch);

186 3              /* printer */
              call write$list(ch);

187 3              /* disk file */
              DO;
188 4                  I=X.BUFPTR;
189 4                  IF I=length(X.BUFFER) THEN$DO
191 5                      CALL FLUSH$BUFFER (PTR);
192 5                      I=0;
193 5                  END$IF;
194 4                  X.BUFFER(I)=CH;
195 4                  X.BUFPTR=I+1;
196 4              END;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

197 3      end$case;
198 2      end outbyte;

199 1      open$input: proc (ptr);
200 2          dcl ptr addr,
                x based ptr file$$structure;

201 2          x.bufptr=length(x.buffer);
202 2          call disk$select(x.disk);
203 2          IF LGW (VERSION) >= 30H THEN$DO
205 3              IF OPEN$RO$FILE (.X.FCBLOCK) <> OFFH THEN RETURN;
                ELSE$DO
209 3                  IF OPEN$FILE (.X.FCBLOCK) <> OFFH THEN RETURN;
                END$IF;
212 2          CALL FILEABORT (.X, .DPENERRTEXT);
213 2      end open$input;

214 1      open$output: proc(ptr);
215 2          dcl ptr addr,
                x based ptr file$$structure;

216 2          if devicetype(x.disk)=disktype then$do
218 3              x.bufptr=0;
219 3              call disk$select(x.disk);
220 3              CALL delete$file(.x.fcbblock);
221 3              if create$file(.x.fcbblock) = Offh then
222 3                  CALL FILEABORT (.X, .MAKEERRTEXT);
                end$if;
224 2      end open$output;

225 1      outputclose: proc(ptr);
226 2          dcl ptr addr,
                x based ptr file$$structure;

227 2          if devicetype(x.disk)=disktype then$do
229 3              call outbyte(end$of$file,.x);
230 3              CALL FLUSH$BUFFER (PTR);
231 3              IF CLOSE$FILE (.X.FCBLOCK) = OFFH THEN
232 3                  CALL FILEABORT (.X, .CLOSEERRTEXT);
                end$if;
234 2      end outputclose;

235 1      INPUT$CLOSE: PROCEDURE (PTR);
236 2          DECLARE PTR ADDRESS, X BASED PTR FILE$$STRUCTURE;
237 2          CALL DISK$SELECT (X.DISK);
238 2          CALL SET$DMA$ADDRESS (.X.BUFFER);
239 2          IF CLOSE$FILE (.X.FCBLOCK) THEN;
241 2      END INPUT$CLOSE;

242 1      outhexbyte: proc(ch) public;
243 2          dcl ch byte;
244 2          call outbyte(ch,.hex$file);
245 2      end outhexbyte;

246 1      outprintbyte: proc(ch) public;
247 2          dcl ch byte;
248 2          if printfile.disk=console then$do

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```
250 3      call write$console(ch);
251 3      else$do
253 3          if error$printed then call write$console(ch);
255 3          call outbyte(ch,.printfile);
256 3      end$if;
257 2      end outprintbyte;
```

```
258 1      outsymbolbyte: proc(ch) public;
259 2          decl ch byte;
260 2          call outbyte(ch,.symbolfile);
261 2      end outsymbolbyte;
```

```
262 1      insourcebyte: proc byte public;
263 2          return inbyte(.sourcefile);
264 2      end insourcebyte;
```

```
265 1      inincludebyte: proc byte public;
266 2          return inbyte(.includefile);
267 2      end inincludebyte;
```

```
268 1      opensource: proc public;
269 2          CALL open$input(.sourcefile);
270 2      end opensource;
```

```
271 1      openinclude: proc public;
272 2          CALL open$input(.includefile);
273 2      end openinclude;
```

```
274 1      openhex: proc public;
275 2          CALL open$output(.hexfile);
276 2      end openhex;
```

```
277 1      openprint: proc public;
278 2          CALL open$output(.printfile);
279 2      end openprint;
```

```
280 1      opensymbol: proc public;
281 2          CALL open$output(.symbolfile);
282 2      end opensymbol;
```

```
283 1      close$source: proc public;
284 2          call input$close (.source$file);
285 2      end close$source;
```

```
286 1      rewindsource: proc public;
287 2          sourcefile.fcbblock(nr)=0;
288 2          sourcefile.bufptr=length(sourcefile.buffer);
289 2          if sourcefile.fcbblock(ex) <> 0 then$do
291 3              sourcefile.fcbblock(ex)=0;
292 3              sourcefile.fcbblock(s2)=0;
293 3              CALL opensource;
294 3          end$if;
295 2      end rewindsource;
```

```
296 1      close$include: proc public;
297 2          call input$close (.include$file);
298 2      end close$include;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

299 1  closehex: proc public;
300 2      call outputclose(.hexfile);
301 2  end closehex;

302 1  closeprint: proc public;
303 2      call outputclose(.printfile);
304 2  end closeprint;

305 1  closesymbol: proc public;
306 2      call outputclose(.symbolfile);
307 2  end closesymbol;

308 1  i$file$setup: proc(dev,filnam,filtyp) public;
309 2      dcl dev byte,(filnam,filtyp) addr;
310 2      call clearcontrol(.includefile,filtyp);
311 2      includefile.disk=dev;
312 2      call copy(8,filnam,.includefile.fcblock(fn));
313 2  end i$file$setup;

314 1  filessetup:      proc      byte public;
315 2      dcl

          ch          byte,          /* pick up character */
          i           byte,          /* counter */
          noleft      byte,          /* no of characters left in tbuff */
          bpt         byte,          /* index of tbuff */
          exitvalue    byte,          /* exitvalue of subroutine */
          flag        byte;          /* program logic flag */

316 2  nextch: proc byte;
317 3      if noleft > 0 then$do
319 4          ch=tbuff(bpt);
320 4          noleft=noleft-1;
321 4          bpt=bpt+1;
322 4      else$do
324 4          ch=cr;
325 4      end$if;
326 3      return ch;
327 3  end nextch;

328 2  getdsk: procedure (p);
329 3      declare p address, dsk based p byte;
330 3      ch=upper(nextch); /* test selected disk drive */
331 3      if letter(ch) then$do
333 4          dsk=ch-"A";
334 4          if dsk > validdisk then
335 4              if dsk < console then
336 4                  exitvalue = false; /* invalid drive */

          else$do
339 4              exitvalue=false;
340 4              noleft=0;
341 4          end$if;
342 3  end getdsk;

343 2  exitvalue=true;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

/* save current disk */
344 2    default$drive,diskunit=interrogate$disk;
        /* enter user selected disk */
345 2    if fcb(dr) <> 0 then$do
347 3        call selectdisk(diskunit:=fcb(dr)-1);
348 3    end$if;

/* clear control blocks */
349 2    call clearcontrol(.sourcefile,.asmdefault);
350 2    call clearcontrol(.hexfile,.hexdefault);
351 2    call clearcontrol(.printfile,.lstdefault);
352 2    call clearcontrol(.symbolfile,.symdefault);
353 2    call copy(8,.fcb(fn),.sourcefile.fcbblock(fn));
354 2    call copy(8,.fcb(fn),.hexfile.fcbblock(fn));
355 2    call copy(8,.fcb(fn),.printfile.fcbblock(fn));
356 2    call copy(8,.fcb(fn),.symbolfile.fcbblock(fn));

357 2    if FCB (FT) <> SPACE then$do    /* pick up specified source file type */
359 3        call copy(3,.fcb(ft),.sourcefile.fcbblock(ft));
360 3    end$if;

/* Move source file name to SOURCENAME */
361 2    CALL FILL (SPACE, LENGTH (SOURCENAME), .SOURCENAME);
362 2    i=0;
363 2    do while i<8 and (sourcename(i):=sourcefile.fcbblock(fn+i)) <> space;
364 3        i=i+1;
365 3    end$while;
366 2    sourcename(i)='.';
367 2    i=i+1;
368 2    call copy(3,.sourcefile.fcbblock(ft),.sourcename(i));

/* Test if file parameters */
369 2    noleft=tbuff(0);
370 2    bpt=1;
371 2    FLAG = FALSE;
372 2    IF FCB16 (1) <> SPACE THEN$DO
374 3        IF FCB16 (1) <> DOLLAR THEN$DO
376 4            EXITVALUE = FALSE;
377 4        ELSE$DO
379 4            DO WHILE (NOLEFT > 0) AND (NEXTCH <> DOLLAR);
380 5                END$WHILE;
381 4            FLAG = TRUE;
382 4        END$IF;
383 3    END$IF;

384 2    if flag then$do

        /* file parameters present - pick them up */

386 3        do while noleft > 0;
387 4            if (ch:=upper(nextch)) <> space then$do

                /* A-parameter */
                IF CH = "A" THEN call getdisk(.sourcefile.disk);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

391 5      /* H-parameter */
        ELSE IF CH = "h" THEN call getdsk(.hexfile.disk);

393 5      /* P-parameter */
        ELSE IF CH = "P" THEN call getdsk(.printfile.disk);

395 5      /* S-parameter */
        ELSE IF CH = "S" THEN call getdsk(.symbolfile.disk);

397 5      /* F-parameter */
        ELSE IF CH = "F" THEN$DO
399 6          if (ch:=upper(nextch)) = "I" then$do
401 7              intel$hex$on=true;
402 7          else$do
404 7              if ch= "D" then$do
406 8                  intel$hex$on=false;
407 8              else$do
409 8                  exitvalue=false;
410 8                  noleft=0;
411 8              endif;
412 7          endif;
413 6      END$IF;

        /* error, no legal parameter */
        ELSE
414 5            DO;
415 6                exitvalue=false;
416 6                noleft=0;
417 6            END$DO;

418 5        end$if;
        end$while;
420 3    end$if;

421 2    printdevice=printfile.disk; /* set global printdevice flag */
422 2    SYMBOLDEVICE = SYMBOLFILE.DISK;
423 2    INCLUDE$DEFAULT = SOURCEFILE.DISK;

        /* input must be from a disk file */

424 2    if devicetype(sourcefile.disk) <> disktype then$do
426 3        exitvalue=false;
427 3    end$if;

428 2    return exitvalue;

429 2    end filesetup;

430 1    end file;

```

CCP/M-86 2.0 V.3

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
106	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(37)
106	0000H	1	ACCLN BYTE EXTERNAL(91)
106	0000H	1	ACCLNSAVE BYTE EXTERNAL(94)
106	0000H	80	ACCUM. BYTE ARRAY(80) EXTERNAL(92)
106	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(93)
2			LITERALLY 61 64 67 70 76 94 97
			100 103 106 109 111 117 140 170 181 200
			215 226 236 309
84	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(31) STACK=0000H
87	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(32) STACK=0000H
105	0000H	1	ASM86TEXT. BYTE ARRAY(1) EXTERNAL(44)
109	0000H	3	ASMDEFAULT BYTE ARRAY(3) DATA 349
106	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
106	0008H	1	BASEINDEX. BYTE MEMBER(COPEPANDS)
3			LITERALLY
3			LITERALLY
315	0EBFH	1	BPT. BYTE 319 321 370
3			LITERALLY
3			LITERALLY
170	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(X) 174
236	0024H	1024	BUFFER BYTE ARRAY(1024) MEMBER(X) 238
109	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(SYMBOLFILE)
226	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(X)
140	0024H	1024	BUFFER BYTE ARRAY(1024) MEMBER(X) 142 146 147
			152 153 165
109	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(PRINTFILE)
200	0024H	1024	BUFFER BYTE ARRAY(1024) MEMBER(X) 201
215	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(X)
109	0024H	1024	BUFFER BYTE ARRAY(1024) MEMBER(INCLUDEFILE)
117	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(X)
109	0024H	1024	BUFFER BYTE ARRAY(1024) MEMBER(SOURCEFILE) 288
109	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(HEXFILE)
181	0024H	512	BUFFER BYTE ARRAY(512) MEMBER(X) 189 194
109	0022H	2	BUFPTR WORD MEMBER(INCLUDEFILE)
109	0022H	2	BUFPTR WORD MEMBER(PRINTFILE)
226	0022H	2	BUFPTR WORD MEMBER(X)
140	0022H	2	BUFPTR WORD MEMBER(X) 141 166
170	0022H	2	BUFPTR WORD MEMBER(X) 173
109	0022H	2	BUFPTR WORD MEMBER(HEXFILE)
215	0022H	2	BUFPTR WORD MEMBER(X) 218
109	0022H	2	BUFPTR WORD MEMBER(SYMBOLFILE)
117	0022H	2	BUFPTR WORD MEMBER(X)
181	0022H	2	BUFPTR WORD MEMBER(X) 188 195
109	0022H	2	BUFPTR WORD MEMBER(SOURCEFILE) 288
236	0022H	2	BUFPTR WORD MEMBER(X)
200	0022H	2	BUFPTR WORD MEMBER(X) 201
81	0000H	1	CH BYTE PARAMETER 82
315	0EBCH	1	CH BYTE 319 324 326 330 331 333 387 389
			391 393 395 397 399 404

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

75	0000H	1	CH	BYTE PARAMETER	76				
246	0004H	1	CH	BYTE PARAMETER AUTOMATIC		247	250	254	255
180	0006H	1	CH	BYTE PARAMETER AUTOMATIC		181	135	165	194
90	0000H	1	CH	BYTE PARAMETER	91				
78	0000H	1	CH	BYTE PARAMETER	79				
242	0004H	1	CH	BYTE PARAMETER AUTOMATIC		243	244		
140	0EBBH	1	CH	BYTE	143 150 165 167				
84	0000H	1	CH	BYTE PARAMETER	85				
121	0004H	1	CH	BYTE PARAMETER AUTOMATIC		122	123	125	127
258	0004H	1	CH	BYTE PARAMETER AUTOMATIC		259	260		
87	0000H	1	CH	BYTE PARAMETER	88				
19	0000H	1	CHAR	BYTE PARAMETER	20				
22	0000H	1	CHAR	BYTE PARAMETER	23				
106	0000H	2	CIP	WORD EXTERNAL(56)					
116	0031H	28	CLEARCONTROL . . .	PROCEDURE STACK=0016H		310	349	350	351
				352					
110	0000H	49	CLEARFCB	PROCEDURE STACK=000EH	118				
105	0000H	1	CLOSEERRTEXT . . .	BYTE ARRAY(1) EXTERNAL(50)		232			
38	0000H		CLOSEFILE	PROCEDURE BYTE EXTERNAL(15) STACK=0000H			231		
				239					
299	0399H	12	CLOSEHEX	PROCEDURE PUBLIC STACK=0020H					
296	038DH	12	CLOSEINCLUDE . . .	PROCEDURE PUBLIC STACK=0012H					
302	03A5H	12	CLOSEPRINT	PROCEDURE PUBLIC STACK=0020H					
283	0361H	12	CLOSESOURCE	PROCEDURE PUBLIC STACK=0012H					
305	0381H	12	CLOSESYMBOL	PROCEDURE PUBLIC STACK=0020H					
106	0000H	1	CODEMACROFLAG . . .	BYTE EXTERNAL(113)					
3			CODEMACROHEAD . . .	LITERALLY					
106	0000H	2	CODEMACROPTR	WORD EXTERNAL(103)					
4			CONSOLE	LITERALLY	125 248 335				
109			CONSOLETYPE	LITERALLY	126				
25	0000H		CONSTAT	PROCEDURE BYTE EXTERNAL(10) STACK=0000H					
66	0000H		COPY	PROCEDURE EXTERNAL(25) STACK=0000H		114	312		
				353 354 355 356 359 368					
2			CR	LITERALLY	324				
50	0000H		CREATEFILE	PROCEDURE BYTE EXTERNAL(19) STACK=0000H			221		
58	0000H		CRLF	PROCEDURE EXTERNAL(22) STACK=0000H					
106	0000H	2	CSCTP	WORD EXTERNAL(61)					
106	0000H	1	CSEGSPEC	BYTE EXTERNAL(59)					
106	0000H	1	CSEGIYPE	BYTE EXTERNAL(57)					
106	0000H	2	CSEGVAlUE	WORD EXTERNAL(58)					
106	0000H	1	CSPEC	BYTE EXTERNAL(68)					
106	0000H	2	CURCSEG	WORD EXTERNAL(64)					
106	0000H	2	CURDSEG	WORD EXTERNAL(65)					
106	0000H	2	CURESEG	WORD EXTERNAL(67)					
106	0000H	9	CURRENTSYMBOL . . .	STRUCTURE EXTERNAL(104)					
106	0000H	2	CURSSEG	WORD EXTERNAL(66)					
96	0000H	2	D	WORD PARAMETER	97				
93	0000H	2	D	WORD PARAMETER	94				
102	0000H	2	D	WORD PARAMETER	103				
66	0000H	2	D	WORD PARAMETER	67				
99	0000H	2	D	WORD PARAMETER	100				
2			DCL	LITERALLY					
102	0000H		DECOUT	PROCEDURE EXTERNAL(37) STACK=0000H					
106	0000H	1	DEFAULTDRIVE	BYTE EXTERNAL(111)	344				
110	0004H	2	DEFAULTPT	WORD PARAMETER AUTOMATIC	111	114			
116	0004H	2	DEFAULTPTR	WORD PARAMETER AUTOMATIC	117	118			
41	0000H		DELETEFILE	PROCEDURE EXTERNAL(16) STACK=0000H			220		

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

106	0001H	1	DESCR.	BYTE MEMBER(TOKEN)				
111	0000H	1	DEST	BYTE BASED(FECP) ARRAY(1)	114			
308	0008H	1	DEV.	BYTE PARAMETER AUTOMATIC	309	311		
121	004DH	39	DEVICETYPE	PROCEDURE BYTE STACK=0004H	182	216	227	
				424				
78	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(29) STACK=0000H				
109	0000H	1	DISK	BYTE MEMBER(PRINTFILE)	248	394	421	
140	0000H	1	DISK	BYTE MEMBER(X)	145			
215	0000H	1	DISK	BYTE MEMBER(X)	216	219		
109	0000H	1	DISK	BYTE MEMBER(HEXFILE)	392			
109	0000H	1	DISK	BYTE MEMBER(SOURCEFILE)	390	423	424	
236	0000H	1	DISK	BYTE MEMBER(X)	237			
200	0000H	1	DISK	BYTE MEMBER(X)	202			
109	0000H	1	DISK	BYTE MEMBER(INCLUDEFILE)	311			
181	0000H	1	DISK	BYTE MEMBER(X)	182			
170	0000H	1	DISK	BYTE MEMBER(X)	171			
117	0000H	1	DISK	BYTE MEMBER(X)	119			
226	0000H	1	DISK	BYTE MEMBER(X)	227			
109	0000H	1	DISK	BYTE MEMBER(SYMBOLFILE)	396	422		
131	0004H	1	DISK	BYTE PARAMETER AUTOMATIC	132	133	135	
29	0000H	1	DISKNUMBER	BYTE PARAMETER	30			
105	0000H	1	DISKREADERRTEXT.	BYTE ARRAY(1) EXTERNAL(47)		156		
131	0074H	23	DISKSELECT	PROCEDURE STACK=0008H	145	171	202	219
				237				
109			DISKTYPE	LITERALLY	129	216	227	424
109	0E8AH	1	DISKUNIT	BYTE	119	133	135	136
105	0000H	1	DISKWRITEERRTXT.	BYTE ARRAY(1) EXTERNAL(48)		176		
55	0000H	2	DMAADDRESS	WORD PARAMETER	56			
109			DOLLAR	LITERALLY	374	379		
109			DR	LITERALLY	345	347		
106	0000H	2	DSCIP.	WORD EXTERNAL(63)				
329	0000H	1	DSK.	BYTE BASED(P)	333	334	335	
106	0000H	1	DSPEC.	BYTE EXTERNAL(69)				
2			ELSEDO	LITERALLY	206	336		
14	0000H	2	ENDBUF	WORD EXTERNAL(5)				
2			ENDCASE.	LITERALLY				
2			ENDDO.	LITERALLY				
2			ENDFOREVER	LITERALLY				
2			ENDIF.	LITERALLY	210	222	232	
2			ENDMODULE.	LITERALLY				
2			ENDOFFILE.	LITERALLY	152	229		
2			ENDPROC.	LITERALLY				
105	0000H	1	ENDTEXT.	BYTE ARRAY(1) EXTERNAL(42)				
2			ENDWHILE	LITERALLY	418			
106	0000H	1	EOFSET	BYTE EXTERNAL(95)				
93	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(34) STACK=0000H				
106	0000H	1	ERRORPRINTED	BYTE EXTERNAL(98)	253			
106	0000H	2	ERRORS	WORD EXTERNAL(88)				
106	0000H	2	ESCTP.	WORD EXTERNAL(60)				
106	0000H	1	ESPEC.	BYTE EXTERNAL(71)				
109			EX	LITERALLY	289	291		
315	0ECON	1	EXITVALUE.	BYTE	336	339	343	376
2			FALSE.	LITERALLY	336	339	371	376
				426				
11	0000H	1	FCB.	BYTE ARRAY(1) EXTERNAL(2)	345	347	353	354
				355	356	357	359	
12	0000H	1	FCB16.	BYTE ARRAY(1) EXTERNAL(3)	372	374		

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

38	0000H	2	FCBADDRESS	WORD PARAMETER	39			
50	0000H	2	FCBADDRESS	WORD PARAMETER	51			
47	0000H	2	FCBADDRESS	WORD PARAMETER	48			
44	0000H	2	FCBADDRESS	WORD PARAMETER	45			
41	0000H	2	FCBADDRESS	WORD PARAMETER	42			
35	0000H	2	FCBADDRESS	WORD PARAMETER	36			
32	0000H	2	FCBADDRESS	WORD PARAMETER	33			
72	0000H	2	FCBADR	WORD PARAMETER	73			
236	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	239			
215	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	220	221		
200	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	205	209		
181	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)				
170	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	175			
140	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	148			
117	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	118			
109	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(SYMBOLFILE)		356		
109	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(PRINTFILE)		355		
109	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(HEXFILE)	354			
109	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(INCLUDEFILE)		312		
109	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(SOURCEFILE)		267	289	
				291 292 353 359 363 368				
226	0001H	33	FCBBLOCK	BYTE ARRAY(33) MEMBER(X)	231			
110	0006H	2	FCBPT.	WORD PARAMETER AUTOMATIC	111	112	113	114
1	0000H		FILE	PROCEDURE STACK=0000H				
72	0000H		FILEABORT.	PROCEDURE EXTERNAL(27) STACK=0000H		156	176	
				212 222 232				
3			FILESTRUCTURE . .	LITERALLY	109 140 200 236			
3			FILEOSTRUCTURE . .	LITERALLY	109 117 170 181	215	226	
314	03E1H	512	FILESETUP.	PROCEDURE BYTE PUBLIC STACK=001AH				
75	0000H		FILL	PROCEDURE EXTERNAL(28) STACK=0000H		112	113	
				361				
308	0006H	2	FILNAM	WORD PARAMETER AUTOMATIC	309	312		
308	0004H	2	FILTYP	WORD PARAMETER AUTOMATIC	309	310		
106	0000H	2	FTKSTNACKOPT. . .	WORD EXTERNAL(99)				
315	0EC1H	1	FLAG	BYTE	371 381 384			
169	0110H	76	FLUSHBUFFER. . . .	PROCEDURE STACK=000EH		191	230	
109			FN	LITERALLY	113 312 353 354 355 356 363			
2			FOREVER.	LITERALLY				
2			FORMFEED	LITERALLY				
109			FT	LITERALLY	114 357 359 368			
106	0000H	1	FULLSYMBTAB. . . .	BYTE EXTERNAL(101)				
8	0000H	1	FUNC	BYTE PARAMETER	9			
5	0000H	1	FUNC	BYTE PARAMETER	6			
328	060CH	68	GETDSK	PROCEDURE STACK=0008H	390	392	394	396
107	0000H		GLOBALINIT	PROCEDURE EXTERNAL(114) STACK=0000H				
106	0000H	5	HELP	BYTE ARRAY(5) EXTERNAL(109)				
96	0000H		HEX1OUT.	PROCEDURE EXTERNAL(35) STACK=0000H				
99	0000H		HEX2OUT.	PROCEDURE EXTERNAL(36) STACK=0000H				
109	0003H	3	HEXDEFAULT	BYTE ARRAY(3) DATA	350			
109	0848H	548	HEXFILE.	STRUCTURE	244 275 300 350 354 392			
315	0EB0H	1	I.	BYTE	362 363 364 366 367 368			
181	0EB8H	2	I.	WORD	188 189 192 194 195			
170	0EB6H	2	I.	WORD	172 173 174 177			
140	0EB4H	2	I.	WORD	141 142 144 146 147 152 153 160			
				163 165 166				
106	0000H	1	I.	BYTE EXTERNAL(110)				
308	03BDH	36	IFILESETUP	PROCEDURE PUBLIC STACK=0020H				

106	0000H	1	IFLEVEL.	BYTE EXTERNAL(103)		
106	0000H	1	IFLIST.	BYTE EXTERNAL(74)		
139	0088H	146	INBYTE.	PROCEDURE BYTE STACK=000EH	253	266
106	0000H		INCLUDEDEFAULT. . .	BYTE EXTERNAL(112)	423	
109	0424H	1060	INCLUDEFILE. . . .	STRUCTURE	266 272 297 310 311 312	
106	0000H	1	INCLUDEON.	BYTE EXTERNAL(102)		
3			INDEXREGBIT. . . .	LITERALLY		
3			INDEXRECCOUNT. . .	LITERALLY		
8	0000H	2	INFO.	WORD PARAMETER	9	
5	0000H	2	INFO.	WORD PARAMETER	6	
265	0319H	12	ININCLUDEBYTE. . .	PROCEDURE BYTE PUBLIC STACK=0012H		
2			INIT.	LITERALLY		
105	0000H	1	INITIALS.	BYTE ARRAY(1) EXTERNAL(40)		
235	0296H	35	INPUTCLOSE. . . .	PROCEDURE STACK=000EH	284 297	
262	0300H	12	INSOURCEBYTE. . .	PROCEDURE BYTE PUBLIC STACK=0012H		
106	0000H	1	INTELHEXON. . . .	BYTE EXTERNAL(96)	401 405	
53	0000H		INTERROGATEDISK. .	PROCEDURE BYTE EXTERNAL(20) STACK=0000H		344
3			IREGBIT.	LITERALLY		
3			IREGCOUNT. . . .	LITERALLY		
			LENGTH.	BUILTIN	142 146 153 189 201 288 361	
106	0000H	2	LENGTH.	WORD MEMBER(OPERANDS)		
106	0000H	2	LENGTH.	WORD MEMBER(CURRENTSYMBOL)		
81	0000H		LETTER.	PROCEDURE BYTE EXTERNAL(30) STACK=0000H		331
2			LF.	LITERALLY		
2			LIT.	LITERALLY	2 3 4 109	
			LOW.	BUILTIN	203	
109	0006H	3	LSTDEFAULT. . . .	BYTE ARRAY(3) DATA	351	
106	0000H	2	MACROPTR.	WORD EXTERNAL(100)		
105	0000H	1	MAKEERRTEXT. . . .	BYTE ARRAY(1) EXTERNAL(49)	222	
106	0000H	1	MAXCOL.	BYTE EXTERNAL(75)		
5	0000H		MON1.	PROCEDURE EXTERNAL(0) STACK=0000H		
8	0000H		MON2.	PROCEDURE BYTE EXTERNAL(1) STACK=0000H		
96	0000H	1	N.	BYTE PARAMETER	97	
102	0000H	2	N.	WORD PARAMETER	103	
99	0000H	2	N.	WORD PARAMETER	100	
93	0000H	1	N.	BYTE PARAMETER	54	
75	0000H	1	N.	BYTE PARAMETER	76	
66	0000H	1	N.	BYTE PARAMETER	67	
63	0000H	2	N.	WORD PARAMETER	64	
106	0000H	1	NEXTCH.	BYTE EXTERNAL(90)		
316	05E1H	43	NEXTCH.	PROCEDURE BYTE STACK=0002H	330 379 387	
				399		
105	0000H	1	NO <.	BYTE EXTERNAL(39)		
106	0000H	1	NOERRDK.	BYTE EXTERNAL(97)		
315	0EBEH	1	NOLEFT.	BYTE	317 320 340 369 379 386 410 416	
106	0000H	1	NOOPER.	BYTE EXTERNAL(106)		
3			NOOVCOUNT. . . .	LITERALLY		
3			NOOVERRIDEBIT. . .	LITERALLY		
109			NR.	LITERALLY	287	
4			NULL.	LITERALLY	123	
109			NULLTYPE.	LITERALLY	124	
106	0006H	2	OFFSET.	WORD MEMBER(OPERANDS)		
106	0006H	2	OFFSET.	WORD MEMBER(CURRENTSYMBOL)		
105	0000H	1	OPENERRTEXT. . . .	BYTE ARRAY(1) EXTERNAL(46)	212	
32	0000H		OPENFILE.	PROCEDURE BYTE EXTERNAL(13) STACK=0000H		269
274	033DH	12	OPENHEX.	PROCEDURE PUBLIC STACK=0012H		
271	0331H	12	OPENINCLUDE. . . .	PROCEDURE PUBLIC STACK=0012H		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

199	01CCH	71	OPENINPUT.	PROCEDURE STACK=000EH	269	272			
214	0213H	72	OPENOUTPUT.	PROCEDURE STACK=000EH	275	278	281		
277	0349H	12	OPENPRINT.	PROCEDURE PUBLIC STACK=0012H					
35	0000H		OPENROFILE.	PROCEDURE BYTE EXTERNAL(14) STACK=0000H				205	
268	0325H	12	OPENSOURCE.	PROCEDURE PUBLIC STACK=0012H	293				
280	0355H	12	OPENSYMBOL.	PROCEDURE PUBLIC STACK=0012H					
106	0000H	36	OPERANDS.	STRUCTURE ARRAY(4) EXTERNAL(107)					
3			OPERANDSTRUC.	LITERALLY 105					
180	0169H	99	OUTBYTE.	PROCEDURE STACK=0016H	229	244	255	260	
242	02BEH	17	OUTHEXBYTE.	PROCEDURE PUBLIC STACK=001CH					
246	02CFH	45	OUTPRINTBYTE.	PROCEDURE PUBLIC STACK=001CH					
225	025BH	64	OUTPUTCLOSE.	PROCEDURE STACK=001CH	300	303	306		
258	02FCH	17	OUTSYMBOLBYTE.	PROCEDURE PUBLIC STACK=001CH					
69	0000H		OUTTEXT.	PROCEDURE EXTERNAL(26) STACK=0000H					
328	0004H	2	P.	WORD PARAMETER AUTOMATIC	329	333	334	335	
106	0000H	1	PAGESIZE.	BYTE EXTERNAL(81)					
105	0000H	1	PAGETEXT.	BYTE ARRAY(1) EXTERNAL(41)					
105	0000H	1	PARERRTEXT.	BYTE ARRAY(1) EXTERNAL(45)					
106	0000H	1	PASS.	BYTE EXTERNAL(55)					
105	0000H	1	PASSOTEXT.	BYTE ARRAY(1) EXTERNAL(53)					
105	0000H	1	PASSTEXT.	BYTE ARRAY(1) EXTERNAL(54)					
116	0006H	2	POINT.	WORD PARAMETER AUTOMATIC	117	118	119		
106	0000H	240	PREFIX.	BYTE ARRAY(240) EXTERNAL(85)					
106	0000H	1	PREFIXPTR.	BYTE EXTERNAL(86)					
106	0000H	1	PRINTDEVICE.	BYTE EXTERNAL(73)	421				
4			PRINTER.	LITERALLY 127					
109			PRINTERTYPE.	LITERALLY 128					
109	0A6CH	548	PRINTFILE.	STRUCTURE 248 255 278 303 351 355 394					
				421					
106	0000H	1	PRINTON.	BYTE EXTERNAL(72)					
106	0000H	1	PRINTSWITCHOFF.	BYTE EXTERNAL(73)					
2			PROC.	LITERALLY 60 63 84 87 96 99 102					
				110 121 139 180 199 214 225 242 246 258					
				262 265 268 271 274 277 280 283 286 296					
				299 302 305 308 314 316					
15	0000H	2	PT.	WORD PARAMETER 76					
235	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	236	237	238	239	
225	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	246	227	229	230	
				231 232					
214	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	215	216	218	219	
				220 221 222					
199	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	200	201	202	205	
				209 212					
180	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	181	182	188	191	
				194 195					
169	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	170	171	173	174	
				175 176					
139	0004H	2	PTR.	WORD PARAMETER AUTOMATIC	140	141	145	147	
				148 152 156 165 166					
17	0000H		READCONSOLE.	PROCEDURE BYTE EXTERNAL(7) STACK=0000H					
44	0000H		READRECORD.	PROCEDURE BYTE EXTERNAL(17) STACK=0000H				148	
2			REENT.	LITERALLY					
286	036DH	32	REWINDSOURCE.	PROCEDURE PUBLIC STACK=0016H					
93	0000H	2	S.	WORD PARAMETER 94					
66	0000H	2	S.	WORD PARAMETER 67					
109			S2.	LITERALLY 292					
106	0000H	12	SAVESOURCE.	BYTE ARRAY(12) EXTERNAL(77)					

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

3		SEGBIT.	LITERALLY						
3		SEGMOUNT.	LITERALLY						
106	0004H	2	SEGMENT.	WORD MEMBER(COPERANDS)					
106	0004H	2	SEGMENT.	WORD MEMBER(CURRENTSYMBOL)					
3		SEGTYPEBAT.	LITERALLY						
3		SEGTYPECOUNT.	LITERALLY						
29	0000H		SELECTDISK.	PROCEDURE EXTERNAL(12) STACK=0000H	155	247			
55	0000H		SETDMAADDRESS.	PROCEDURE EXTERNAL(21) STACK=0000H	147	174			
				238					
106	0003H	1	SFLAG.	BYTE MEMBER(COPERANDS)					
106	0003H	1	SFLAG.	BYTE MEMBER(CURRENTSYMBOL)					
106	0000H	1	SIFORM.	BYTE EXTERNAL(82)					
106	0000H	80	SOURCEBUF.	BYTE ARRAY(80) EXTERNAL(83)					
109	0000H	1060	SOURCEFILE.	STRUCTURE	263	269	284	287	288
					289	291			
					292	349	353	359	363
					368	390	423	424	
106	0000H	12	SOURCENAME.	BYTE ARRAY(12) EXTERNAL(76)	361	363	366		
				368					
106	0000H	1	SOURCEPTR.	BYTE EXTERNAL(84)					
2			SPACE.	LITERALLY	357	361	363	372	387
106	0000H	2	SSCIP.	WORD EXTERNAL(62)					
106	0000H	1	SSPEC.	BYTE EXTERNAL(70)					
2			STRUC.	LITERALLY	106	109	117	140	170
					215	226	236		
106	0002H	1	STYPE.	BYTE MEMBER(COPERANDS)					
106	0002H	1	STYPE.	BYTE MEMBER(CURRENTSYMBOL)					
105	0000H	1	SYMBFULLERRTEXT.	BYTE ARRAY(1) EXTERNAL(51)					
106	0000H	1	SYMBOLDEVICE.	BYTE EXTERNAL(79)			422		
109	0C90H	548	SYMBOLFILE.	STRUCTURE	260	281	306	352	356
					396	422			
3			SYMBOLHEAD.	LITERALLY					
3			SYMBOLSTRUC.	LITERALLY	106				
106	0000H	2	SYMBTAGADR.	WORD EXTERNAL(105)					
109	0009H	3	SYMDEFAULT.	BYTE ARRAY(3) DATA	352				
15	0000H		SYSTEMRESET.	PROCEDURE EXTERNAL(6) STACK=0000H					
69	0000H	2	T.	WORD PARAMETER	70				
2			TAB.	LITERALLY					
13	0000H	1	TRUFF.	BYTE ARRAY(1) EXTERNAL(4)	319	369			
72	0000H	2	TEXTADR.	WORD PARAMETER	73				
2			THENDO.	LITERALLY	133	142	148	150	189
					203	215			
					227	248	289	317	331
					345	357	372	374	384
					387	397	399	404	424
106	0000H	30	TITLE.	BYTE ARRAY(30) EXTERNAL(80)					
106	0000H	4	TOKEN.	STRUCTURE EXTERNAL(89)					
2			TRUE.	LITERALLY	343	381	401		
106	0000H	1	TYPE.	BYTE MEMBER(TOKEN)					
3			TYPEBT.	LITERALLY					
60	0000H		TYPECALC.	PROCEDURE BYTE EXTERNAL(23) STACK=0000H					
3			TYPECOUNT.	LITERALLY					
90	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(33) STACK=0000H					
					387	399			
105	0000H	1	USBREAKTEXT.	BYTE ARRAY(1) EXTERNAL(52)					
105	0000H	1	USEFACTOR.	BYTE ARRAY(1) EXTERNAL(43)					
60	0000H	2	VAL.	WORD PARAMETER	61				
4			VALTODISK.	LITERALLY	334				
106	0002H	2	VALUE.	WORD MEMBER(TOKEN)					
27	0000H		VERSION.	PROCEDURE WORD EXTERNAL(11) STACK=0000H					
63	0000H		WRDTEST.	PROCEDURE BYTE EXTERNAL(24) STACK=0000H					
19	0000H		WRITECONSOLE.	PROCEDURE EXTERNAL(8) STACK=0000H	185	250			

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

22	0000H		WRITELIST.	254	PROCEDURE EXTERNAL(C9) STACK=0000H	185
47	0000H		WRITERECORD.		PROCEDURE BYTE EXTERNAL(C9) STACK=0000H	175
236	0000H	1060	X.		STRUCTURE BASED(PTR)	237 238 239
226	0000H	548	X.		STRUCTURE BASED(PTR)	227 229 231 232
215	0000H	548	X.		STRUCTURE BASED(PTR)	216 218 219 220 221
				222		
200	0000H	1060	X.		STRUCTURE BASED(PTR)	201 202 205 209 212
181	0000H	548	X.		STRUCTURE BASED(PTR)	182 188 189 194 195
170	0000H	548	X.		STRUCTURE BASED(PTR)	171 173 174 175 176
140	0000H	1060	X.		STRUCTURE BASED(PTR)	141 142 145 146 147
				148 152 153 156 165 166		
117	0000H	548	X.		STRUCTURE BASED(PTR)	118 119
105	0000H	1	YES.		BYTE EXTERNAL(C8)	

MODULE INFORMATION:

CODE AREA SIZE = 0650H 16160
 CONSTANT AREA SIZE = 000CH 120
 VARIABLE AREA SIZE = 0EC2H 37780
 MAXIMUM STACK SIZE = 0020H 320
 947 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____



ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE SCANN

OBJECT MODULE PLACED IN SCAN.OBJ

COMPILER INVOKED BY: :FO: SCAN.PLM DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XXX

```

1      $title ('SCANNER MODULE')
      scanm:
      do;

      /*

      modified 3/26/81 R. Silberstein
      modified 3/30/81 R. Silberstein
      modified 4/10/81 R. Silberstein
      modified 9/2/81 R. Silberstein

      */

      $include (:f1:macro.lit)
=      $nolist
      $include (:f1:struct.lit)
=      $nolist
      $include (:f1>equals.lit)
=      $nolist
      $include (:f1:files.ext)
=      $nolist
      $include (:f1:predef.ext)
=      $nolist
      $include (:f1:subr2.ext)
=      $nolist
      $include (:f1:print.ext)
=      $nolist
      $include (:f1:global.ext)
=      $nolist

      /* Variables : */

107 1      dcl
      eofound      byte,      /* true if end-of-file is found */
      lowercase    byte,      /* false if stringinput, otherwise true */
      crfound      byte,      /* true if previous input was CR */
      printready   byte,      /* true if output line to be printed */
      stacksave    addr;      /* save of stack pointer */

      /* Routine to perform unnormal exit from module */

108 1      exit: proc;
109 2      stackptr=stacksave; /* restore input stack */
110 2      end exit;

      /* Put printcharacter into printfile output buffer */

111 1      putprintchar: proc(ch);
112 2      dcl ch byte;
113 2      sourcebuf(sourceptr)=ch;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

114 2      if sourceptr < last(sourcebuf) then$do
116 3          sourceptr=sourceptr+1;
117 3      endif;
118 2      end putprintchar;

/* Read single character from input file. Put characters
   except CR-LF to printbuffer. Convert to uppercase letters */

119 1      read$input: proc byte;
120 2          dcl ch byte;
121 2          if eof$found then call exit; /* unnormal exit */
                                           /* read byte from file */
123 2          if include$on then ch=ininclude$byte; else ch=insource$byte;
126 2          if ch=end$of$file then$do /* test for end-of-file */
128 3              eof$found=true;
129 3          else$do
131 3              if cr$found and ch=lf then$do /* ignore LF after CR */
133 4                  ch=space;
134 4              else$do;
137 4                  if ch=cr then$do /* test for CR */
139 5                      cr$found=true;
140 5                  else$do;
143 5                      cr$found=false;
144 5                      call putprintchar(ch);
145 5                      if ch=lf then ch=space; /* interpret LF within line as space */
                        endif;
148 4                  endif;
149 3              endif;
150 2          if not lowercase then$do /* convert to uppercase */
152 3              ch=upper(ch);
153 3          endif;
154 2          return ch;
155 2      end read$input;

/* skip blanks and tab's in input */

156 1      skip$blanks: proc;
157 2          do while nextch=space or nextch=tab;
158 3              nextch=read$input;
159 3          end$while;
160 2      end skip$blanks;

/* Put character into accumulator */

161 1      putaccum: proc(ch);
162 2          dcl ch byte;
163 2          accum(acclen)=ch;
164 2          if acclen < last(accum) then$do
166 3              acclen=acclen+1;
167 3          endif;
168 2      end put$accum;

```

```

/* Routine to scan remainder of token until a non-

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93350

SER. # _____

alphanumeric character is found. Skip blanks
behind token */

```

169 1  get$remainder: proc(num);
170 2      dcl (cont,num) byte;
171 2      cont=true;
172 2      do while cont;
173 3          do while alphanumeric(nextch:=read$input);
174 4              call putaccum(nextch);
175 4          end$while;
176 3          cont=false;
177 3          if nextch = "0" or nextch = "_" then$do
179 4              cont=true;
180 4              if numb then call putaccum(nextch);
              end$if;
          end$while;
184 2      call skipblanks;
185 2  end get$remainder;

```

/* Routine to scan a text string. Called from SCAN */

```

186 1  stringr: proc;
187 2      dcl cont byte;
188 2      lowercase=true;
189 2      acclen=0;
190 2      cont=true;
191 2      do while cont;
192 3          nextch=read$input;
193 3          do while nextch <> "" and nextch <> cr;
194 4              call putaccum(nextch);
195 4              nextch=read$input;
196 4          end$while;
197 3          if nextch="" then$do
199 4              if (nextch:=read$input) = "" then$do /* interpret "" as ' */
201 5                  call putaccum(nextch);
202 5              else$do
204 5                  lowercase=false;
205 5                  call skipblanks;
206 5                  token.type=string;
207 5                  cont=false;
208 5              end$if;
209 4          else$do
211 4              lowercase=false;
212 4              token.type=error;
213 4              cont=false;
214 4          end$if;
215 3          end$while;
216 2  end stringr;

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

/* Routine to scan a number. Called from SCAN. Test syntax
of number, compute binary value. */

```

217 1  numbr: proc;
218 2      dcl
          nobase      byte, /* number system, 2,8,10 or 16 */
          maxlen      byte, /* max legal no of digits */

```

```

(i,j)      byte, /* counters */
ch         byte,
value      addr, /* 16 bit binary value */
errorflag  byte; /* syntax error flag */

219  2      errorflag=false;
220  2      call getremainder(true); /* get rest of token */
221  2      ch=accum(acclen-1); /* pick up last character of token */
222  2      j=acclen-2;

/* 0 (binary) */
223  2      IF CH = "0" THEN
224  2          do; nobase=2; maxlgth=16; end;

/* 0 or Q (octal) */
228  2      ELSE IF CH = "0" OR CH = "Q" THEN
229  2          do; nobase=8; maxlgth=6; end;

/* H (hexadecimal) */
233  2      ELSE IF CH = "H" THEN
234  2          do; nobase=16; maxlgth=4; end;

/* D (decimal) */
238  2      ELSE IF CH = "D" THEN
239  2          do; nobase=10; maxlgth=5; end;

/* no subscript, default=decimal */
ELSE
243  2      do; nobase=10; maxlgth=5; j=j+1; end;

248  2      i=0ffh; /* skip leading zeros */
249  2      do while accum(i:=i+1) = "0"; end;
251  2      if j < maxlgth+1 then$do
253  3          value=0; /* syntax check number, compute binary value */
254  3          do while i <= j;
255  4              ch=accum(i);
256  4              ch=ch-"0";
257  4              if ch > 9 then ch=ch-7;
259  4              if ch >= nobase then$do
261  5                  errorflag=true;
262  5                  end$if;
263  4              value=value*nobase+ch;
264  4              i=i+1;
265  4              end$while;
266  3          else$do
268  3              errorflag=true;
269  3              end$if;

270  2      if errorflag then$do
272  3          token.type=error;
273  3      else$do
275  3          token.type=number;
276  3          token.descr=0;
277  3          token.value=value;
278  3      end$if;

279  2      end numbr;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
/* Routine to scan an identifier. Lookup identifier in table
for predefined symbols */
```

```
280 1  identr: proc;
281 2    call get$remainder(false); /* get rest of token into accumulator */
      /* look up identifier */
282 2    if not pfind(acclen,.accum(0),.token) then$do
284 3      token.type=ident;
285 3    end$if;
286 2  end identr;
```

```
/* PUBLIC subroutines : */
```

```
287 1  scaninit: proc public;
288 2    eofset,eoffound,crfound,lowercase,printready=false;
289 2    CALL FILL (SPACE, SIZE (PREFIX), .PREFIX);
290 2    CALL FILL (SPACE, LENGTH (ABSADDR), .ABSADDR);
291 2    sourceptr,prefixptr=0;
292 2    call printinit; /* initiate print module */
293 2    call rewindsource;
294 2    nextch=space;
295 2  end scaninit;
```

```
296 1  scan: proc public;
297 2    stacksave=stackptr;
298 2    if printready then$do
300 3      call print$source$line;
301 3      print$ready=false;
302 3    end$if;
303 2    call skipblanks;
304 2    if eoffound then$do
306 3      token.type=spec;
307 3      if crfound then$do
309 4        eoffound=false;
310 4        eofset=true;
311 4      else$do
313 4        printready=true; /* terminate line before EOF */
314 4        crfound=true;
315 4        accum(0)=cr;
316 4      end$if;
317 3    else$do
319 3      acclen=1;
320 3      accum(0)=nextch;

      /* identifier */
321 3      IF LETTER (NEXTCH) THEN call identr;

      /* number */
323 3      ELSE IF DIGIT (NEXTCH) THEN call numbr;

      /* string */
325 3      ELSE IF NEXTCH = '"' THEN call stringr;

      /* special letter */
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

```

ELSE
327 3      do;
328 4      token.type=spec;
329 4      if nextch="I" then accum(0) = cr;
331 4      IF NEXTCH = ";" THEN$DO
333 5          DO WHILE ACCUM (0) <> CR;
334 6              ACCUM (0) = READINPUT;
335 6          END$WHILE;
336 5      END$IF;
337 4      nextch=space;
338 4      if crfound then$do
340 5          print$ready=true;
341 5      else$do
343 5          call skipblanks;
344 5      end$if;
345 4      end;

346 3      end$if;
347 2      end scan;

348 1      skip$rest$of$line: proc public;
349 2          do while accum(0) <> cr;
350 3              call scan;
351 3          end$while;
352 2      end skip$rest$of$line;

353 1      specialtoken: proc(tok) byte public;
354 2          dcl tok byte;
355 2          if (token.type=spec) and (accum(0)=tok) then return true;
357 2          return false;
358 2      end specialtoken;

359 1      skip$until: proc(tok) byte public;
360 2          dcl tok byte;
361 2          do forever;
362 3              if token.type=spec then$do
364 4                  if accum(0)=tok then$do
366 5                      call scan;
367 5                      return true;
368 5                  end$if;
369 4                  if accum(0)=cr then return false;
                      end$if;
372 3                  call scan;
373 3              end$forever;
374 2          end skip$until;

375 1      emptyline: proc byte public;
376 2          return specialtoken(cr);
377 2      end emptyline;

378 1      end$module scanm;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
49	0000H	2	A. WORD PARAMETER 50
104	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(70) 290
104	0000H	1	ACCLEN BYTE EXTERNAL(74) 163 164 165 169 221
			222 282 319
104	0000H	1	ACCLENSAVE BYTE EXTERNAL(77)
104	0000H	80	ACCUM. BYTE ARRAY(80) EXTERNAL(75) 163 164 221
			249 255 282 315 320 330 333 334 349 355
			364 369
104	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(76)
2			LITERALLY 45 53 59 77 80 83 86
			104 107 218
67	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(24) STACK=0000H 173
70	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(25) STACK=0000H
104	0008H	1	BASEINDEX. BYTE MEMBER(OPLRANDS)
104	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
3			BASEREGBIT LITERALLY
3			BASEREGCOUNT LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
5			BYT. LITERALLY
9	0000H	1	CH BYTE PARAMETER 10
218	000FH	1	CH BYTE 221 223 228 233 238 255 256 257
			258 259 263
120	0008H	1	CH BYTE 124 125 126 131 133 137 144 145
			146 152 154
88	0000H	1	CH BYTE PARAMETER 89
111	0004H	1	CH BYTE PARAMETER AUTOMATIC 112 113
70	0000H	1	CH BYTE PARAMETER 71
73	0000H	1	CH BYTE PARAMETER 74
161	0004H	1	CH BYTE PARAMETER AUTOMATIC 162 163
67	0000H	1	CH BYTE PARAMETER 68
64	0000H	1	CH BYTE PARAMETER 65
61	0000H	1	CH BYTE PARAMETER 62
15	0000H	1	CH BYTE PARAMETER 16
58	0000H	1	CH BYTE PARAMETER 59
12	0000H	1	CH BYTE PARAMETER 13
104	0000H	2	CIP. WORD EXTERNAL(39)
38	0000H		CLOSEHEX PROCEDURE EXTERNAL(13) STACK=0000H
36	0000H		CLOSEINCLUDE PROCEDURE EXTERNAL(12) STACK=0000H
40	0000H		CLOSEPRINT PROCEDURE EXTERNAL(14) STACK=0000H
32	0000H		CLOSESOURCE. PROCEDURE EXTERNAL(10) STACK=0000H
42	0000H		CLOSESYMBOL. PROCEDURE EXTERNAL(15) STACK=0000H
4			CODE LITERALLY
104	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(96)
3			CODEMACROHEAD. LITERALLY
104	0000H	2	CODEMACROPTR WORD EXTERNAL(91)
170	0009H	1	CONT BYTE 171 172 176 179
187	000AH	1	CONT BYTE 190 191 207 213
2			CR LITERALLY 137 193 315 330 333 349 369

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

107 0006H 1 CRFOUND. . . . . 376
104 0000H 2 CSCIP. . . . . BYTE 131 139 143 288 307 314 328
104 0000H 1 CSEGSPEC . . . . . WORD EXTERNAL(44)
104 0000H 1 CSEGTYP. . . . . BYTE EXTERNAL(42)
104 0000H 2 CSEGVALU. . . . . BYTE EXTERNAL(40)
104 0000H 1 CSPEC. . . . . WORD EXTERNAL(41)
104 0000H 2 CURCSEG. . . . . BYTE EXTERNAL(51)
104 0000H 2 CURDSEG. . . . . WORD EXTERNAL(47)
104 0000H 2 CURESEG. . . . . WORD EXTERNAL(48)
104 0000H 9 CURRENTSYMBOL. . . . . WORD EXTERNAL(50)
104 0000H 2 CURSSEG. . . . . STRUCTURE EXTERNAL(87)
76 0000H 2 D. . . . . WORD EXTERNAL(49)
79 0000H 2 D. . . . . WORD PARAMETER 77
85 0000H 2 D. . . . . WORD PARAMETER 80
82 0000H 2 D. . . . . WORD PARAMETER 86
2 DCL. . . . . WORD PARAMETER 83
85 0000H DECOU. . . . . LITERALLY
104 0000H 1 DEFAULTDRIVE . . . . . PROCEDURE EXTERNAL(30) STACK=0000H
4 DELETEDSYMB. . . . . BYTE EXTERNAL(94)
104 0001H 1 DESCR. . . . . LITERALLY
44 0000H 1 DEV. . . . . BYTE MEMBER(TOKEN) 276
61 0000H 4 DIGIT. . . . . BYTE PARAMETER 45
104 0000H 2 DOUBLEDEFINED. . . . . PROCEDURE BYTE EXTERNAL(22) STACK=0000H 323
104 0000H 2 DSCIP. . . . . LITERALLY
104 0000H 1 DSPEC. . . . . WORD EXTERNAL(46)
5 DWRD. . . . . BYTE EXTERNAL(52)
102 0000H EJECT. . . . . LITERALLY
2 ELSEDN. . . . . PROCEDURE EXTERNAL(37) STACK=0000H
375 045FH 11 EMPTYLINE. . . . . LITERALLY
2 ENDCASE. . . . . PROCEDURE BYTE PUBLIC STACK=0000H
2 ENDDO. . . . . LITERALLY
2 ENDFOREVER. . . . . LITERALLY
2 ENDIF. . . . . LITERALLY
2 ENDMODULE. . . . . LITERALLY
2 ENDOFFILE. . . . . LITERALLY
2 ENDPROC. . . . . LITERALLY
2 ENDWHILE. . . . . LITERALLY
107 0004H 1 EOFFOUND. . . . . 146 181 370
104 0000H 1 EOFSET. . . . . LITERALLY
76 0000H 4 EQUAL. . . . . LITERALLY
218 0010H 1 ERRORFLAG. . . . . 126
104 0000H 1 ERRORPRINTED. . . . . LITERALLY
104 0000H 2 ERRORS. . . . . LITERALLY
104 0000H 2 ESCIP. . . . . 182
104 0000H 1 ESPEC. . . . . BYTE 121 128 288 304 309
108 0000H 10 EXIT. . . . . BYTE EXTERNAL(78) 288 310
2 FALSE. . . . . PROCEDURE BYTE EXTERNAL(27) STACK=0000H
55 0000H 2 FCBADR. . . . . LITERALLY
55 0000H 3 FILEABORT. . . . . 212 272
3 FILESTRUCTURE. . . . . BYTE 219 261 268 270
3 FILEDSTRUCTURE. . . . . BYTE EXTERNAL(81)
47 0000H 3 FILESETUP. . . . . WORD EXTERNAL(71)
58 0000H 1 FILL. . . . . WORD EXTERNAL(43)
44 0000H 2 FILNAM. . . . . BYTE EXTERNAL(54)
PROCEDURE STACK=0002H 122
LITERALLY 143 176 204 207 211 213 219
281 288 301 309 357 370
WORD PARAMETER 56
PROCEDURE EXTERNAL(20) STACK=0000H
LITERALLY
LITERALLY
PROCEDURE BYTE EXTERNAL(17) STACK=0000H
PROCEDURE EXTERNAL(21) STACK=0000H 289 290
WORD PARAMETER 45

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

44	0000H	2	FILTP	WORD PARAMETER	45	
104	0000H	2	FIRSTMACROPTR . . .	WORD EXTERNAL(82)		
2			FOREVER	LITERALLY	361	
2			FORMFEED	LITERALLY		
104	0000H	1	FULLSYMBTAB	BYTE EXTERNAL(84)		
169	00E6H	85	GETREMATNDR	PROCEDURE STACK=0012H	220	231
105	0000H		GLOBALINIT	PROCEDURE EXTERNAL(97) STACK=0000H		
104	0000H	5	HELP	BYTE ARRAY(5) EXTERNAL(92)		
79	0000H		HEX1OUT	PROCEDURE EXTERNAL(28) STACK=0000H		
82	0000H		HEX2OUT	PROCEDURE EXTERNAL(29) STACK=0000H		
104	0000H	1	I	BYTE EXTERNAL(93)		
218	0000H	1	I	BYTE	248 249 251 254 255 264	
4			IDENT	LITERALLY	284	
280	02DDH	37	IDENTR	PROCEDURE STACK=0016H	322	
44	0000H		IFILESETUP	PROCEDURE EXTERNAL(16) STACK=0000H		
104	0000H	1	IFLEVEL	BYTE EXTERNAL(86)		
104	0000H	1	IFLIST	BYTE EXTERNAL(57)		
104	0000H	1	INCLUDEDFAULT . . .	BYTE EXTERNAL(95)		
104	0000H	1	INCLUDEON	BYTE EXTERNAL(85)	123	
3			INDEXREGBIT	LITERALLY		
3			INDEXREGCOUNT . . .	LITERALLY		
20	0000H		ININCLUDEBYTE . . .	PROCEDURE BYTE EXTERNAL(4) STACK=0000H		124
2			INIT	LITERALLY		
18	0000H		INSOURCEBYTE	PROCEDURE BYTE EXTERNAL(3) STACK=0000H		125
104	0000H	1	INTELHEXON	BYTE EXTERNAL(79)		
3			IREGBIT	LITERALLY		
3			IREGCOUNT	LITERALLY		
218	000EH	1	J	BYTE	222 246 251 254	
4			LAB	LITERALLY		
			LAST	BUILTIN	114 164	
8			LEFTBRACKET	LITERALLY		
			LENGTH	BUILTIN	290	
104	0000H	2	LENGTH	WORD MEMBER(OPERANDS)		
104	0000H	2	LENGTH	WORD MEMBER(CURRENTSYMBOL)		
64	0000H		LETTER	PROCEDURE BYTE EXTERNAL(23) STACK=0000H		321
2			LF	LITERALLY	131 145	
2			LIT	LITERALLY	2 3 4 5 6 7 8	
107	0005H	1	LOWERCASE	BYTE	150 188 204 211 288	
104	0000H	2	MACROPTR	WORD EXTERNAL(83)		
104	0000H	1	MAXCOL	BYTE EXTERNAL(58)		
218	000CH	1	MAXLGTH	BYTE	226 231 236 241 245 251	
76	0000H	1	N	BYTE PARAMETER	77	
58	0000H	1	N	BYTE PARAMETER	59	
49	0000H	1	N	BYTE PARAMETER	50	
85	0000H	2	N	WORD PARAMETER	86	
82	0000H	2	N	WORD PARAMETER	83	
79	0000H	1	N	BYTE PARAMETER	80	
4			NEGLECTED	LITERALLY		
104	0000H	1	NEXTCH	BYTE EXTERNAL(73)	157 158 173 174 177	
					181 192 193 194 195 197 199 201 294 320	
					321 323 325 329 331 337	
5			NIL	LITERALLY		
218	000BH	1	NOBASE	BYTE	225 230 235 240 244 259 263	
104	0000H	1	NOERROR	BYTE EXTERNAL(80)		
104	0000H	1	NOOPER	BYTE EXTERNAL(89)		
3			NOOVERCOUNT	LITERALLY		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

3			NOOVERRTDEBIT. . .	LITERALLY	
169	0004H	1	NUMB	BYTE PARAMETER AUTOMATIC	170 180
4			NUMBER	LITERALLY 275	
217	018BH	290	NUMBR.	PROCEDURE STACK=0016H	324
8			OAND	LITERALLY	
8			OEQ.	LITERALLY	
104	0006H	2	OFFSET	WORD MEMBER(CPERANDS)	
104	0006H	2	OFFSET	WORD MEMBER(CURRENTSYMBOL)	
8			OGE.	LITERALLY	
8			OGT.	LITERALLY	
8			OLAST.	LITERALLY	
8			OLE.	LITERALLY	
8			OLENGTH.	LITERALLY	
8			OLT.	LITERALLY	
8			OMOD	LITERALLY	
8			ONE.	LITERALLY	
8			ONOT	LITERALLY	
8			OOFFSET.	LITERALLY	
8			ODR.	LITERALLY	
26	0000H		OPENHEX.	PROCEDURE EXTERNAL(7) STACK=0000H	
24	0000H		OPENINCLUDE.	PROCEDURE EXTERNAL(6) STACK=0000H	
28	0000H		OPENPRINT.	PROCEDURE EXTERNAL(8) STACK=0000H	
22	0000H		OPENSOURCE.	PROCEDURE EXTERNAL(5) STACK=0000H	
30	0000H		OPENSYMBOL.	PROCEDURE EXTERNAL(9) STACK=0000H	
104	0000H	36	OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(90)	
3			OPERANDSTRUC.	LITERALLY 104	
4			OPERATOR	LITERALLY	
8			OPTR	LITERALLY	
8			OSEG	LITERALLY	
8			OSHL	LITERALLY	
8			OSHORT	LITERALLY	
8			OSHR	LITERALLY	
8			OTYPE.	LITERALLY	
9	0000H		OUTHEXBYTE.	PROCEDURE EXTERNAL(0) STACK=0000H	
12	0000H		OUTPRINTBYTE.	PROCEDURE EXTERNAL(1) STACK=0000H	
15	0000H		OUTSYMBOLBYTE.	PROCEDURE EXTERNAL(2) STACK=0000H	
52	0000H		OUTTEXT.	PROCEDURE EXTERNAL(19) STACK=0000H	
8			OXOR	LITERALLY	
104	0000H	1	PAGEIZE.	BYTE EXTERNAL(64)	
104	0000H	1	PASS	BYTE EXTERNAL(38)	
7			PCODEMACRO.	LITERALLY	
7			PCSEG.	LITERALLY	
7			PDB.	LITERALLY	
7			PDBIT.	LITERALLY	
7			PDD.	LITERALLY	
7			PDSEG.	LITERALLY	
7			PDW.	LITERALLY	
7			PEJECT.	LITERALLY	
7			PEND.	LITERALLY	
7			PENDIF.	LITERALLY	
7			PENDM.	LITERALLY	
7			PEQU.	LITERALLY	
7			PESEG.	LITERALLY	
49	0000H		PFIND.	PROCEDURE BYTE EXTERNAL(18) STACK=0000H	282
7			PIF.	LITERALLY	
7			PIFLIST.	LITERALLY	
7			PINCLUDE.	LITERALLY	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

[illegible]

CCP/M-86 20 V.3
COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # CC-104

4		REG.	LITERALLY				
6		RES.	LITERALLY				
34	0000H	REWTNDSOURCE . . .	PROCEDURE EXTERNAL(11) STACK=0000H	293			
8		RIGHTBRACKET . . .	LITERALLY				
6		RSI.	LITERALLY				
6		RSP.	LITERALLY				
6		RSS.	LITERALLY				
76	0000H	2 S.	WORD PARAMETER 77				
49	0000H	2 S.	WORD PARAMETER 50				
104	0000H	12 SAVESOURCE . . .	BYTE ARRAY(12) EXTERNAL(60)				
296	0348H	189 SCAN	PROCEDURE PUBLIC STACK=001AH	350	366	372	
287	0302H	70 SCANINIT	PROCEDURE PUBLIC STACK=000AH				
1	0000H	SCANM.	PROCEDURE STACK=0000H				
3		SEGMBIT.	LITERALLY				
3		SEGMCOUNT.	LITERALLY				
104	0004H	2 SEGMENT.	WORD MEMBER(COPLRANDS)				
104	0004H	2 SEGMENT.	WORD MEMBER(CURRENTSYMBOL)				
3		SEGTYPEBIT	LITERALLY				
3		SEGTYPECOUNT . . .	LITERALLY				
104	0003H	1 SFLAG.	BYTE MEMBER(COPLRANDS)				
104	0003H	1 SFLAG.	BYTE MEMBER(CURRENTSYMBOL)				
104	0000H	1 SIMFORM.	BYTE EXTERNAL(65)				
		SIZE	BUILTIN 289				
156	00A0H	41 SKIPBLANKS	PROCEDURE STACK=000CH	184	205	303	343
348	0405H	17 SKIPPRESTOFLINE . .	PROCEDURE PUBLIC STACK=001EH				
359	0432H	45 SKIPUNTIL	PROCEDURE BYTE PUBLIC STACK=0020H				
104	0000H	80 SOURCEBUF.	BYTE ARRAY(80) EXTERNAL(66)	113	114		
104	0000H	12 SOURCENAME	BYTE ARRAY(12) EXTERNAL(59)				
104	0000H	1 SOURCEPTR.	BYTE EXTERNAL(67)	113	114	116	291
2		SPACE.	LITERALLY 133 146 157 289	290	294	337	
4		SPEC.	LITERALLY 306 328 355 362				
353	0416H	28 SPECIALTOKEN . . .	PROCEDURE BYTE PUBLIC STACK=0004H	376			
104	0000H	2 SSCTP.	WORD EXTERNAL(45)				
104	0000H	1 SSPEC.	BYTE EXTERNAL(53)				
		STACKPTR	BUILTIN 109 297				
107	0000H	2 STACKSAVE.	WORD 109 297				
4		STRING.	LITERALLY 206				
186	0138H	128 STRINGR.	PROCEDURE STACK=0010H	326			
2		STRUC.	LITERALLY 104				
104	0002H	1 STYPE.	BYTE MEMBER(COPLRANDS)				
104	0002H	1 STYPE.	BYTE MEMBER(CURRENTSYMBOL)				
4		SYMBOL.	LITERALLY				
104	0000H	1 SYMBOLDEVICE	BYTE EXTERNAL(62)				
3		SYMBOLHEAD	LITERALLY				
3		SYMBOLSTRUC.	LITERALLY 104				
104	0000H	2 SYMBTABADR	WORD EXTERNAL(88)				
52	0000H	2 T.	WORD PARAMETER 53				
2		TAB.	LITERALLY 157				
55	0000H	2 TEXTADR.	WORD PARAMETER 56				
2		THENDO	LITERALLY 114 126 131 137 150 164 177				
			197 199 251 259 270 282 298 304 307 331				
			338 362 364				
104	0000H	30 TITLE.	BYTE ARRAY(30) EXTERNAL(63)				
359	0004H	1 TOK.	BYTE PARAMETER AUTOMATIC	360	354		
353	0004H	1 TOK.	BYTE PARAMETER AUTOMATIC	354	355		
104	0000H	4 TOKEN.	STRUCTURE EXTERNAL(72)	206	212	272	275
			276 277 282 284 306 328 355 352				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 573

PACIFIC GROVE, CA 93950

SER. # _____

2	TRUE	LITERALLY	128	139	171	179	188	190	233
			251	268	310	313	314	340	356
104	0000H	1 TYPE	BYTE MEMBER(TOKEN)		206	212	272	275	284
			306	328	355	362			
3	TYPEBIT.	LITERALLY							
3	TYPECOUNT.	LITERALLY							
4	UDFSYMB	LITERALLY							
73	0000H	UPPER.	PROCEDURE BYTE EXTERNAL(26) STACK=0000H						152
99	0000H	1 USEFACT.	BYTE PARAMETER	100					
218	0002H	2 VALUE.	WORD	253	263	277			
104	0002H	2 VALUE.	WORD MEMBER(TOKEN)		277				
4	VARIABLE	LITERALLY							
5	WRD.	LITERALLY							

MODULE INFORMATION:

CODE AREA SIZE	= 046AH	1130D
CONSTANT AREA SIZE	= 0000H	0D
VARIABLE AREA SIZE	= 0011H	17D
MAXIMUM STACK SIZE	= 0020H	32D
848 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____



ISIS-II PL/M-86 V2.0 COMPIATION OF MODULE PRINT

OBJECT MODULE PLACED IN PRINT.OBJ

COMPILER INVOKED BY: :F0: PRINT.PLH DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

```

1      $title ("PRINT MODULE")
      print:
      do;

/*

      modified 3/26/81 R. Silberstein
      modified 3/30/81 R. Silberstein
      modified 4/7/81 R. Silberstein
      modified 4/9/81 R. Silberstein
      modified 4/16/81 R. Silberstein
      modified 4/20/81 R. Silberstein
      modified 5/5/81 R. Silberstein
      modified 7/24/81 R. Silberstein
      modified 7/27/81 R. Silberstein
      modified 8/19/81 R. Silberstein
      modified 9/2/81 R. Silberstein
      modified 9/19/81 R. Silberstein

*/

      $include (:f1:macro.lit)
= $nolist
      $include (:f1:struc.lit)
= $nolist
      $INCLUDE (:F1:DEV.LIT)
= $nolist
      $include (:f1:files.ext)
= $nolist
      $include (:f1:subr2.ext)
= $nolist
      $include (:f1:global.ext)
= $nolist
      $include (:f1:text.ext)
= $nolist

85 1    dcl
      pageno      byte, /* current page no */
      lineno      byte, /* current line no */
      col         byte, /* column counter */
      field1start lit "6", /* start of hexoutput print */
      FIELD15START LIT "19", /* START OF ABSOLUTE ADDRESS FIELD */
      field2start lit "24"; /* start of source output print */

86 1    printbyt: proc(ch);
87 2      dcl ch byte;
88 2      if not ascii(char(ch)) then ch="#";
90 2      if ch <> lf then col=col+1;
92 2      if ch = cr then col=0;
94 2      call outprintbyte(ch);
95 2      end printbyt;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

96  1      advance: proc(n);          /* advance to column "n" */
97  2          dcl n byte;
98  2          do while n > col;
99  3              call printbyt(space);
100 3          end$while;
101 2      end advance;

102 1      printtext: proc(s);
103 2          dcl s address,ch based s byte;
104 2          DO WHILE CH <> 0;
105 3              CALL PRINTBYT (CH);
106 3              S = S + 1;
107 3          END;
108 2      end printtext;

109 1      printhead: proc;
110 2          COL = 0;
111 2          pageno=pagenu+1;
112 2          call printtext(.initials);
113 2          call printtext(.sourcename);
114 2          call printtext(.( " ",0));
115 2          call printtext(.title);
116 2          call advance(maxcol-11);
117 2          call printtext(.pagetext);
118 2          call decout(pagenu,.help(0));
119 2          call printtext(.help(1));
120 2          call printtext(.(cr,lf,cr,lf,cr,lf,0));
121 2          lineno=4;
122 2      end printhead;

      /* Public routine to perform page eject */

123 1      eject: proc public;
124 2          if simform then$do
126 3              do while (lineno:=lineno+1) <= pagesize;
127 4                  call printbyt(cr);
128 4                  call printbyt(lf);
129 4              end$while;
130 3          else$do
132 3              call outprintbyte(formfeed);
133 3          end$if;
134 2          lineno=0;
135 2      end eject;

136 1      printnewpage: proc public;
137 2          IF LINENO > 4 THEN$DO
139 3              call eject;
140 3              call printhead;
141 3          END$IF;
142 2      end printnewpage;

143 1      incrementline: proc;
144 2          lineno = lineno + 1;
145 2          if lineno >= pagesize - 10 then call printnewpage;
147 2      end incrementline;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```
/* Print single byte, update column counter,
   expand tabs (each 8.th column) */
```

```
148 1  print$single$byte: proc(ch) public;
149 2  dcl ch byte;
150 2  if ch=tab then$do
151 3    ch=8-((col-field1start) mod 8);
152 3    do while (ch:=ch-1) <> 0ffh;
153 4      call printbyt(space);
154 4    end$while;
155 3  else$do
156 3    call printbyt(ch);
157 3    if ch = lf then call incrementline;
158 3  end$if;
159 2  end print$single$byte;

162 2

163 1  print$CrLf: proc public;
164 2  call print$single$byte(cr);
165 2  call print$single$byte(lf);
166 2  end print$CrLf;
```

```
/* Print a field given by last column of field, source-
   array containing ascii bytes, index of this array, and
   index of last byte of source array. Before entry, the
   current column position must be start of this field. */
```

```
167 1  print$field: proc (sourceindex,s,lastindex,stopcol);
168 2  dcl (sourceindex,s,lastindex) address,
      stopcol byte,
      source based s (1) byte,
      k based sourceindex byte,
      last based lastindex byte;
```

```
169 2  do while col < stopcol and k < last;
170 3    call print$single$byte(source(k));
171 3    k=k+1;
172 3  end$while;
173 2  end print$field;
```

```
174 1  print$sl: proc;
175 2  dcl (i,j) byte;
176 2  DECLARE K BYTE;
```

```
177 2  IF (PRINTDEVICE = NULL) AND NOT ERRORPRINTED THEN RETURN; /* NO NEED TO WASTE TIME
178 2  HERE */
179 2  if include$on then$do
180 3    prefix(0)=" ";
181 3    if prefixptr=0 then prefixptr=1;
182 3  end$if;
183 2  i,j,col=0;
184 2  /* print first field of line prefix */
185 2  call printfield(.i,.prefix(0),.prefixptr,field1start);
```

```
/* Print rest of prefix and source.
   If line overflow, print rest on
   following lines. */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

187 2      if prefixptr-i+sourceptr > 0 then$do
189 3          do while (prefixptr-i) + (sourceptr-j) > 0;
190 4              call advance(field1start);
191 4              call printfield(
192 4                  .i,.prefix(0),.prefixptr,((field1start-1)/3)*3);
194 5              IF ABSADDR (0) <> SPACE THEN$DO
195 5                  CALL ADVANCE (FIELD15START);
196 6                  DO K = 0 TO 3;
197 6                      CALL PRINTSINGLEBYTE (ABSADDR (K));
198 5                  END$IF;
199 4              if sourceptr-j > 0 then$do
201 5                  call advance(field2start);
202 5                  call printfield(.j,.sourcebuf(0),.sourceptr,maxcol-1);
203 5              end$if;
204 4              call printcrlf;
205 4          end$while;
206 3      else$do
208 3          call printcrlf;
209 3      end$if;
210 2      end print$sl;

/* Public routine to print prefix and source line on printfile. */

211 1      print$source$line: proc public;
212 2          IF PRINT$ON OR ERRORPRINTED THEN CALL PRINT$SL;
214 2          CALL FILL (SPACE, PREFIXPTR, .PREFIX);
215 2          CALL FILL (SPACE, LENGTH (ABSADDR), .ABSADDR);
216 2          prefixptr,sourceptr=0;
217 2      end print$source$line;

/* Public routine to initiate print module */

218 1      printinit: proc public;
219 2          if print$on then$do
221 3              pageno=0;
222 3              LINEND = OFFH;
223 3              CALL PRINTNEWPAGE;
224 3          end$if;
225 2      end printinit;

/* Public routine to print module information on printfile */

226 1      printterminate: proc (USEFACT) public;
227 2          DECLARE USEFACT BYTE;
228 2          if print$on then$do
230 3              CALL PRINTCRLF;
231 3              CALL PRINTCRLF;
232 3              call printtext(.endtext); /* END OF ASSEMBLY. NO OF ERRORS: */
233 3              call decout(errors,.help(0));
234 3              call printtext(.help(2));
235 3              CALL PRINTTEXT (.USEFACT0R);
236 3              CALL DECOU (USEFACT, .HELP (0));
237 3              CALL PRINTTEXT (.HELP (3));
238 3              CALL PRINTTEXT (.(25H,CR,LF,0)); /* & */
239 3          end$if;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

240 2 end printterminate;

241 1 endmodule print;

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
81	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(62) 192 196 215
81	0000H	1	ACCLEN. BYTE EXTERNAL(66)
81	0000H	1	ACCLENSAVE. BYTE EXTERNAL(69)
81	0000H	80	ACCUH. BYTE ARRAY(80) EXTERNAL(67)
81	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(68)
2			LITERALLY 41 46 52 70 73 76 79
			81
96	0032H	24	ADVANCE. PROCEDURE STACK=000EH 116 190 194 201
60	0000H		ALPHANUMERIC. PROCEDURE BYTE EXTERNAL(23) STACK=0000H
63	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(24) STACK=0000H 89
84	0000H	1	ASM86TEXT. BYTE ARRAY(1) EXTERNAL(96)
81	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
81	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
3			BASEREGBIT. LITERALLY
3			BASEREGCOUNT. LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
148	0004H	1	CH. BYTE PARAMETER AUTOMATIC 149 150 152 153
			158 159
54	0000H	1	CH. BYTE PARAMETER 55
5	0000H	1	CH. BYTE PARAMETER 6
103	0000H	1	CH. BYTE BASED(CS) 104 105
57	0000H	1	CH. BYTE PARAMETER 58
8	0000H	1	CH. BYTE PARAMETER 9
66	0000H	1	CH. BYTE PARAMETER 67
86	0004H	1	CH. BYTE PARAMETER AUTOMATIC 87 88 89 90
			92 94
63	0000H	1	CH. BYTE PARAMETER 64
60	0000H	1	CH. BYTE PARAMETER 61
51	0000H	1	CH. BYTE PARAMETER 52
11	0000H	1	CH. BYTE PARAMETER 12
81	0000H	2	CIP. WORD EXTERNAL(31)
84	0000H	1	CLOSEERRTEXT. BYTE ARRAY(1) EXTERNAL(102)
34	0000H		CLOSEHEX. PROCEDURE EXTERNAL(13) STACK=0000H
32	0000H		CLOSEINCLUDE. PROCEDURE EXTERNAL(12) STACK=0000H
36	0000H		CLOSEPRINT. PROCEDURE EXTERNAL(14) STACK=0000H
28	0000H		CLOSESOURCE. PROCEDURE EXTERNAL(10) STACK=0000H
38	0000H		CLOSESYMBOL. PROCEDURE EXTERNAL(15) STACK=0000H
81	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(88)
3			CODEMACROHEAD. LITERALLY
81	0000H	2	CODEMACROPTR. WORD EXTERNAL(83)
85	0002H	1	COL. BYTE 91 93 98 110 152 169 185
4			CONSOLE. LITERALLY
2			CR. LITERALLY 92 120 127 164 238
81	0000H	2	CSCTP. WORD EXTERNAL(36)
81	0000H	1	CSEGSPEC. BYTE EXTERNAL(34)
81	0000H	1	CSECTYPE. BYTE EXTERNAL(32)
81	0000H	2	CSEGVAlUE. WORD EXTERNAL(33)

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

81	0000H	1	CSPEC.	BYTE EXTERNAL(43)	
81	0000H	2	CURCSEG.	WORD EXTERNAL(39)	
81	0000H	2	CUPDSEG.	WORD EXTERNAL(40)	
81	0000H	2	CURKSEG.	WORD EXTERNAL(42)	
81	0000H	9	CURRENTSYMBOL. . .	STRUCTURE EXTERNAL(79)	
81	0000H	2	CURSSEG.	WORD EXTERNAL(41)	
69	0000H	2	D.	WORD PARAMETER	70
72	0000H	2	D.	WORD PARAMETER	73
75	0000H	2	D.	WORD PARAMETER	76
78	0000H	2	D.	WORD PARAMETER	79
2			DCL.	LITERALLY	
78	0000H		DECOUT	PROCEDURE EXTERNAL(29) STACK=0000H	118 233
				236	
81	0000H	1	DEFAULTORIVE . . .	BYTE EXTERNAL(86)	
81	0001H	1	DESCR.	BYTE MEMBER(10KEN)	
40	0000H	1	DEV.	BYTE PARAMETER	41
54	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(21) STACK=0000H	
84	0000H	1	DISKREADERRTEXT. .	BYTE ARRAY(1) EXTERNAL(99)	
84	0000H	1	DISKWRITEERRXT. .	BYTE ARRAY(1) EXTERNAL(100)	
81	0000H	2	DSCIP.	WORD EXTERNAL(38)	
81	0000H	1	DSPEC.	BYTE EXTERNAL(44)	
123	0000H	51	EJECT.	PROCEDURE PUBLIC STACK=0000H	139
2			ELSEDD.	LITERALLY	
2			ENDCASE.	LITERALLY	
2			ENDDO.	LITERALLY	
2			ENDFOREVER.	LITERALLY	
2			ENDIF.	LITERALLY	160 183
2			ENDMODULE.	LITERALLY	
2			ENDOFFILE.	LITERALLY	
2			ENDPROC.	LITERALLY	
84	0000H	1	ENDTEXT.	BYTE ARRAY(1) EXTERNAL(94)	232
2			ENDWHILE.	LITERALLY	
81	0000H	1	EOFSET.	BYTE EXTERNAL(70)	
69	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(26) STACK=0000H	
81	0000H	1	ERRORPRINTED. . .	BYTE EXTERNAL(73)	177 212
81	0000H	2	ERRORS.	WORD EXTERNAL(63)	233
81	0000H	2	ESCAP.	WORD EXTERNAL(35)	
81	0000H	1	ESPEC.	BYTE EXTERNAL(46)	
2			FALSE.	LITERALLY	
48	0000H	2	FCBADR.	WORD PARAMETER	49
85			FIELD15START. . . .	LITERALLY	191 194
85			FIELD1START. . . .	LITERALLY	186 190
85			FIELD2START. . . .	LITERALLY	152 201
48	0000H		FILEABORT.	PROCEDURE EXTERNAL(19) STACK=0000H	
3			FILESTRUCTURE. . .	LITERALLY	
3			FILEOSTRUCTURE. . .	LITERALLY	
43	0000H		FILESETUP.	PROCEDURE BYTE EXTERNAL(17) STACK=0000H	
51	0000H		FILL.	PROCEDURE EXTERNAL(20) STACK=0000H	214 215
40	0000H	2	FILNAM.	WORD PARAMETER	41
40	0000H	2	FILTYP.	WORD PARAMETER	41
81	0000H	2	FIRSTINACROPTR. . .	WORD EXTERNAL(74)	
2			FOREVER.	LITERALLY	
2			FORMFEED.	LITERALLY	132
81	0000H	1	FULLSYMBTAB. . . .	BYTE EXTERNAL(76)	
82	0000H		GLOBALINIT.	PROCEDURE EXTERNAL(39) STACK=0000H	
81	0000H	5	HELP.	BYTE ARRAY(5) EXTERNAL(84)	118 119 233
				234 236 237	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

72	0000H		HEX1OUT.	PROCEDURE EXTERNAL(27) STACK=0000H
75	0000H		HEX2OUT.	PROCEDURE EXTERNAL(28) STACK=0000H
81	0000H	1	I.	BYTE EXTERNAL(85)
175	0003H	1	I.	BYTE 135 136 137 139 191
40	0000H		IFILESETUP.	PROCEDURE EXTERNAL(16) STACK=0000H
81	0000H	1	IFLEVEL.	BYTE EXTERNAL(78)
81	0000H	1	IFLIST.	BYTE EXTERNAL(49)
81	0000H	1	INCLUDEDEFAULT.	BYTE EXTERNAL(87)
81	0000H	1	INCLUDEON.	BYTE EXTERNAL(77)
143	0102H	27	INCREMENTLINE.	PROCEDURE STACK=001AH 179 160
3			INDEXREGBIT.	LITERALLY
3			INDEXREGCOUNT.	LITERALLY
16	0000H		INCLUDEBYTE.	PROCEDURE BYTE EXTERNAL(4) STACK=0000H
2			INIT.	LITERALLY
84	0000H	1	INITIALS.	BYTE ARRAY(1) EXTERNAL(92) 112
14	0000H		INSOURCEBYTE.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H
81	0000H	1	INTELHEXON.	BYTE EXTERNAL(71)
3			IREGBIT.	LITERALLY
3			IREGCOUNT.	LITERALLY
175	0004H	1	J.	BYTE 185 189 199 202
176	0005H	1	K.	BYTE 195 196
168	0000H	1	K.	BYTE BASED(SOURCEINDEX) 169 170 171
168	0000H	1	LAST.	BYTE BASED(LASINDEX) 169
167	0006H	2	LASTINDEX.	WORD PARAMETER AUTOMATIC 168 169
81	0000H	2	LENGTH.	WORD MEMBER(CURRENTSYMBOL)
			LENGTH.	BUILTIN 215
81	0000H	2	LENGTH.	WORD MEMBER(OPERANDS)
57	0000H		LETTER.	PROCEDURE BYTE EXTERNAL(22) STACK=0000H
2			LF.	LITERALLY 90 120 128 159 165 238
85	0001H	1	LTHERO.	BYTE 121 126 134 137 144 145 222
2			LIT.	LITERALLY 2 3 4 85
81	0000H	2	MACROPTR.	WORD EXTERNAL(75)
84	0000H	1	MAKEERRTEXT.	BYTE ARRAY(1) EXTERNAL(101)
81	0000H	1	MAXCOL.	BYTE EXTERNAL(50) 116 202
51	0000H	1	N.	BYTE PARAMETER 52
78	0000H	2	N.	WORD PARAMETER 79
75	0000H	2	N.	WORD PARAMETER 76
72	0000H	1	N.	BYTE PARAMETER 73
69	0000H	1	N.	BYTE PARAMETER 70
96	0004H	1	N.	BYTE PARAMETER AUTOMATIC 97 98
81	0000H	1	NEXTCH.	BYTE EXTERNAL(65)
84	0000H	1	NO.	BYTE EXTERNAL(91)
81	0000H	1	NOERROR.	BYTE EXTERNAL(72)
81	0000H	1	NOOPER.	BYTE EXTERNAL(81)
3			NOOVERCOUNT.	LITERALLY
3			NOOVERRIDEBIT.	LITERALLY
4			NULL.	LITERALLY 177
81	0006H	2	OFFSET.	WORD MEMBER(OPERANDS)
81	0006H	2	OFFSET.	WORD MEMBER(CURRENTSYMBOL)
84	0000H	1	OPENERRTEXT.	BYTE ARRAY(1) EXTERNAL(98)
22	0000H		OPENHEX.	PROCEDURE EXTERNAL(7) STACK=0000H
20	0000H		OPENINCLUDE.	PROCEDURE EXTERNAL(6) STACK=0000H
24	0000H		OPENPRINT.	PROCEDURE EXTERNAL(8) STACK=0000H
18	0000H		OPENSOURCE.	PROCEDURE EXTERNAL(5) STACK=0000H
26	0000H		OPENSYMBOL.	PROCEDURE EXTERNAL(9) STACK=0000H
81	0000H	36	OPERANDS.	STRUCTURE ARRAY(4) EXTERNAL(82)
3			OPERANDSTRUC.	LITERALLY 81

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

5	0000H		OUTHEXBYTE	PROCEDURE EXTERNAL(C0) STACK=0000H					
8	0000H		OUTPRINTBYTE	PROCEDURE EXTERNAL(C1) STACK=0000H	94	132			
11	0000H		OUTSYMBOLEBYTE	PROCEDURE EXTERNAL(C2) STACK=0000H					
45	0000H		OUTTEXT	PROCEDURE EXTERNAL(C10) STACK=0000H					
85	0000H	1	PAGENO	BYTE 111 118 221					
81	0000H	1	PAGESTZE	BYTE EXTERNAL(C55) 125 145					
84	0000H	1	PAGETEXT	BYTE ARRAY(C1) EXTERNAL(C93)	117				
84	0000H	1	PAKERTEXT	BYTE ARRAY(C1) EXTERNAL(C97)					
91	0000H	1	PASS	BYTE EXTERNAL(C30)					
84	0000H	1	PASSTEXT	BYTE ARRAY(C1) EXTERNAL(C105)					
84	0000H	1	PASSITEXT	BYTE ARRAY(C1) EXTERNAL(C106)					
81	0000H	240	PREFIX	BYTE ARRAY(240) EXTERNAL(C60)	181	186	191		
				214					
81	0000H	1	PREFIXPTR	BYTE EXTERNAL(C61) 182 183 186 187 189					
				191 214 216					
1	0000H		PRINT	PROCEDURE STACK=0000H					
86	0000H	50	PRINTBYT	PROCEDURE STACK=0008H 154 158	99	105	127	129	
163	0160H	17	PRINTCRLF	PROCEDURE PUBLIC STACK=0024H 231	204	208	230		
81	0000H	1	PRINTDEVICE	BYTE EXTERNAL(C53) 177					
4			PRINTER	LITERALLY					
167	0171H	60	PRINTFIELD	PROCEDURE STACK=002CH 186 191 202					
109	0063H	90	PRINTHEADER	PROCEDURE STACK=0012H 140					
218	02CAH	25	PRINTINIT	PROCEDURE PUBLIC STACK=0014H					
136	00F0H	18	PRINTNLWPAGE	PROCEDURE PUBLIC STACK=0016H 146 223					
81	0000H	1	PRINTON	BYTE EXTERNAL(C47) 212 219 228					
148	011DH	67	PRINTSINGLEBYTE . .	PROCEDURE PUBLIC STACK=0020H 164 165 170					
				196					
174	01ADH	229	PRINTSL	PROCEDURE STACK=0030H 213					
211	0292H	56	PRINTSOURCELINE . .	PROCEDURE PUBLIC STACK=0034H					
81	0000H	1	PRINTSWITCHOFF . . .	BYTE EXTERNAL(C48)					
226	02E3H	79	PRINTTERMINATE . . .	PROCEDURE PUBLIC STACK=002AH					
102	004AH	25	PRINTTEXT	PROCEDURE STACK=000EH 112 113 114 115					
				117 119 120 232 234 235 237 238					
2			PROC	LITERALLY 5 8 11 14 16 18 20					
				22 24 26 28 30 32 34 36 38 40					
				43 60 63 72 75 78 86 96 102 109					
				123 136 143 148 163 167 174 211 218 226					
51	0000H	2	PT	WORD PARAMETER 52					
2			REENT	LITERALLY					
30	0000H		REWINDSOURCE	PROCEDURE EXTERNAL(C11) STACK=0000H					
69	0000H	2	S	WORD PARAMETER 70					
167	0008H	2	S	WORD PARAMETER AUTOMATIC 168 170					
102	0004H	2	S	WORD PARAMETER AUTOMATIC 103 104 105 106					
81	0000H	12	SAVESOURCE	BYTE ARRAY(12) EXTERNAL(C52)					
3			SEGMBIT	LITERALLY					
3			SEGMCOUNT	LITERALLY					
81	0004H	2	SEGMENT	WORD MEMBER(COPERANDS)					
81	0004H	2	SEGMENT	WORD MEMBER(CURRENTSYMBOL)					
3			SEGTYPEBIT	LITERALLY					
3			SEGTYPECOUNT	LITERALLY					
81	0003H	1	SFLAG	BYTE MEMBER(COPERANDS)					
81	0003H	1	SFLAG	BYTE MEMBER(CURRENTSYMBOL)					
81	0000H	1	SIFORM	BYTE EXTERNAL(C57) 124					
168	0000H	1	SOURCE	BYTE BASED(CS) ARRAY(C1) 170					

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

81	0000H	80	SOURCEBUF.	BYTE ARRAY(30) EXTERNAL(58)	202			
167	000AH	2	SOURCEINDEX.	WORD PARAMETER AUTOMATIC	168	169	170	171
81	0000H	12	SOURCENAME.	BYTE ARRAY(12) EXTERNAL(51)				
81	0000H	1	SOURCEPTR.	BYTE EXTERNAL(59)	187	189	199	202 216
2			SPACE.	LITERALLY	99	154	192	214 215
81	0000H	2	SSCIP.	WORD EXTERNAL(37)				
81	0000H	1	SSPEC.	BYTE EXTERNAL(45)				
167	0004H	1	STOPCOL.	BYTE PARAMETER AUTOMATIC	168	169		
2			STRUC.	LITERALLY	81			
81	0002H	1	STYPE.	BYTE MEMBER(OPERANDS)				
81	0002H	1	STYPE.	BYTE MEMBER(CURRENTSYMBOL)				
84	0000H	1	SYMBFULLERRTEXT.	BYTE ARRAY(1) EXTERNAL(103)				
81	0000H	1	SYMBOLDEVICE.	BYTE EXTERNAL(54)				
3			SYMBOLHEAD.	LITERALLY				
3			SYMBOLSTRUC.	LITERALLY	81			
81	0000H	2	SYMBTABADR.	WORD EXTERNAL(80)				
45	0000H	2	T.	WORD PARAMETER	46			
2			TAB.	LITERALLY	150			
48	0000H	2	TEXTADR.	WORD PARAMETER	49			
2			THEEND.	LITERALLY	124	137	150	179 187 192 199
					219	228		
81	0000H	30	TITLE.	BYTE ARRAY(30) EXTERNAL(55)				115
81	0000H	4	TOKEN.	STRUCTURE EXTERNAL(64)				
2			TRUE.	LITERALLY				
81	0000H	1	TYPE.	BYTE MEMBER(TOKEN)				
3			TYPEBIT.	LITERALLY				
3			TYPECOUNT.	LITERALLY				
66	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(25) STACK=0000H				
84	0000H	1	USBREKTEXT.	BYTE ARRAY(1) EXTERNAL(104)				
226	0004H	1	USLFACT.	BYTE PARAMETER AUTOMATIC	227	236		
84	0000H	1	USEFACTOR.	BYTE ARRAY(1) EXTERNAL(95)				235
4			VALIDDISK.	LITERALLY				
81	0002H	2	VALUE.	WORD MEMBER(TOKEN)				
84	0000H	1	YES.	BYTE EXTERNAL(90)				

MODULE INFORMATION:

CODE AREA SIZE = 0332H 8180
 CONSTANT AREA SIZE = 000EH 140
 VARIABLE AREA SIZE = 0006H 60
 MAXIMUM STACK SIZE = 0034H 520
 615 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

ISIS-II PL/M-86 V2.0 COMPIATION OF MODULE PREDEF

OBJECT MODULE PLACED IN PREDEF.OBJ

COMPILER INVOKED BY: :F0: PREDEF.PLX DEBUG OPTIMIZE(2) DATE(10/5/81) PAGERWIDTH(100) XREF

```

1      $title ("Predefined symbols")
      predef:
      do;

      /*

      modified 7/24/81 R. Silberstein

      */

      /***** Module for predefined symbols *****/

      /*

      This module contains the tables and subroutines
      for the PREDEFINED symbols of the ASM86 assembler.

      The subroutine

          PFIND (nochar,stringaddr,attributeaddr) byte

      defines the interface to the other modules.

      The routine tests if a given symbol is a predefined
      symbol. If so the address of the symbol attributes
      is returned.
      The format of the symbol attributes is :

          byte
          0      * symbol type *
          1      * symbol description *
          2      *
          3      * symbol value *
          *
          *
          *

      */

      /* include language macros */
      /* and general assembler */
      /* definitions (literals). */

      $include(:f1:macro.lit)
= $nolist
      $include(:f1:equals.lit)
= $nolist

      /*      Predefined numbers:      */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
8 1 dcl
    nbyte lit '1';
    nword lit '2';
    ndword lit '4';
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

\$eject

/*

here are the predefined symbols of
the assembler.

The symbols are grouped according to
the symbol lengths. Moreover, each group
is sorted alphabetically so that binary
search algorithm could be used.

*/

```

9 1 dcl
tok0(*) byte data(0), /* symbollength > 1 only */
tok1(*) byte data(0),
tok2(*) byte data(34,"AHALAXBHBLBPBXCHCLCSCXDRDDDDHDIOL",
/*DSOWDXEQESGEGTIFLELTLNLRRAPSRW$ISPSS"),
tok3(*) byte data(11,"ANDENDEQUHDDNOTOROPTRSEGSILSHRXQR"),
tok4(*) byte data(13,"BYTECSEGD$ITDSEGEWNESEGLASTLISTRELRLW",
/*SSEGTPEWJRD"),
tok5(*) byte data(6,"DWORDEJECTENOTIFMODRM$HJRTITLE"),
tok6(*) byte data(5,"IFLISTLENGTHNOLISTOFF$ETSEGFIX"),
tok7(*) byte data(2,"INCLUDESIMFORM"),
tok8(*) byte data(3,"NOIFLISTNOSEGFIXPAGESIZE"),
tok9(*) byte data(2,"CODEMACROPAGEWIDTH");

```

/* Pointer table: */

```

10 1 dcl
tokpointer(*) address data(.tok0,.tok1,.tok2,.tok3,.tok4,
/* .tok5,.tok6,.tok7,.tok8,.tok9);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$object
```

```
/*
```

```
This is the attribute table for
the predefined symbols.
```

```
*/
```

```
11 1
```

```
dcl
```

```
value0
```

```
byte,
```

```
value1
```

```
byte,
```

```
value2(34)
```

```
struc
```

```
(type byte,descr byte,value addr) data(
```

```
reg,byt,rah, reg,byt,ral, /* registers AH and AL */
```

```
reg,wrdrax, reg,byt,rbh, /* registers AX and BH */
```

```
reg,byt,rbl, reg,wrdrbp, /* registers BL and BP */
```

```
reg,wrdrbx, reg,byt,rch, /* registers CX and CH */
```

```
reg,byt,rcl, reg,dwrdr,cs, /* registers CL and CS */
```

```
reg,wrdr,rcx, /* register CX */
```

```
pseudo,nil,pdb, pseudo,nil,pdd, /* pseudos DB and DD */
```

```
reg,byt,rdh, reg,wrdr,rdi, /* registers DH and DI */
```

```
reg,byt,rdl, reg,dwrdr,ds, /* registers DL and DS */
```

```
pseudo,nil,pdw, /* pseudo DW */
```

```
reg,wrdr,rdx, /* register DX */
```

```
operator,nil,oeq, /* operator EQ */
```

```
reg,dwrdr,es, /* register ES */
```

```
operator,nil,oge, /* operator GE */
```

```
operator,nil,ogt, /* operator GT */
```

```
pseudo,nil,pif, /* pseudo IF */
```

```
operator,nil,ole, /* operator LE */
```

```
operator,nil,olt, /* operator LT */
```

```
operator,nil,one, /* operator NE */
```

```
operator,nil,oor, /* operator OR */
```

```
pseudo,nil,prb, /* pseudo RB */
```

```
pseudo,nil,prs, /* pseudo RS */
```

```
pseudo,nil,prw, /* pseudo RW */
```

```
reg,wrdr,rsi, reg,wrdr,rsp, /* registers SI and SP */
```

```
reg,dwrdr,rss, /* register SS */
```

```
value3(11) struc (type byte,descr byte,value addr) data(
```

```
operator,nil,oand, /* operator AND */
```

```
pseudo,nil,pend,pseudo,nil,pequ, /* pseudos END and EQU */
```

```
operator,nil,omod, /* operator MOD */
```

```
operator,nil,onot, /* operator NOT */
```

```
pseudo,nil,porg, /* pseudo ORG */
```

```
operator,nil,optr, /* operator PTR */
```

```
operator,nil,oseg, /* operator SEG */
```

```
operator,nil,oshl, /* operator SHL */
```

```
operator,nil,oshr, /* operator SHR */
```

```
operator,nil,oxor, /* operator XOR */
```

```
value4(13) struc (type byte,descr byte,value addr) data(
```

```
number,byt,nbyte, /* 8 bit number BYTE (1) */
```

```
pseudo,nil,pcseg, /* pseudo CSEG */
```

```
pseudo,nil,pdbit, /* pseudo DBIT */
```

```
pseudo,nil,pdseg, /* pseudo DSEG */
```

```
pseudo,nil,pendm, /* pseudo ENDM */
```

```
pseudo,nil,pseg, /* pseudo ESEG */
```

```
operator,nil,olast, /* operator LAST */
```

```
pseudo,nil,plist, /* pseudo LIST */
```

```
pseudo,nil,prelb, /* pseudo RELB */
```

```
pseudo,nil,prelw, /* pseudo RELW */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

pseudo,nil,pseq, /* pseudo SEQ */
operator,nil,otype, /* operator TYPE */
number,word,word, /* 16 bit number WORD (2) */
value5(6) struct (type byte,descr byte,value addr) data(
number,dword,dword, /* 32 bit number DWORD (4) */
pseudo,nil,pject, /* pseudo JECT */
pseudo,nil,pendif, /* pseudo ENDEF */
pseudo,nil,pmodrm, /* pseudo MODRM */
operator,nil,oshort, /* operator SHORT */
pseudo,nil,ptitle, /* pseudo TITLE */
value6(5) struct (type byte,descr byte,value addr) data(
PSEUDO,NIL,PFLIST, /* PSEUDO IFLIST */
operator,nil,olength, /* operator LENGTH */
pseudo,nil,pnolist, /* pseudo NOLIST */
operator,nil,ooffset, /* operator OFFSET */
pseudo,nil,psegfix, /* pseudo SEGFIX */
value7(2) struct (type byte,descr byte,value addr) data(
pseudo,nil,pinclude, /* pseudo INCLUDE */
pseudo,nil,psimform, /* pseudo SIMFORM */
value8(3) struct (type byte,descr byte,value addr) data(
PSEUDO,NIL,PNOIFLIST, /* PSEUDO NOIFLIST */
pseudo,nil,pnosegfix, /* pseudo NOSEGFIX */
pseudo,nil,ppagesize, /* pseudo PAGESIZE */
value9(2) struct (type byte,descr byte,value addr) data(
pseudo,nil,pcodemacro, /* pseudo CODEMACRO */
pseudo,nil,ppagewidth); /* pseudo PAGEWIDTH */

```

```
/* Pointer table: */
```

12 1

```

dcl
valuepointer(*) address data(.value0,.value1,.value2,.value3,.value4,
                             .value5,.value6,.value7,.value8,.value9);

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```
$object
```

```
/* Global variables and subroutines */
```

```
13 1  dcl
      nochar                byte,
      stringaddr            address,
      attributeaddr         address,
      source based stringaddr (1) byte,
      dest based attributeaddr (1) byte,
      value                 address, /* pointer to attributes */
      valuebyte based value (1) byte,
      tok                   address, /* pointer to table strings */
      tokenbyte based tok   byte,
      t$lookahead           address, /* table string pointer */
      look based t$lookahead (1) byte,
      v$lookahead           address, /* attribute table pointer */
      noleft                byte, /* no of tablestrings left */
      half                  byte, /* noleft/2 */
      i                      byte, /* counter */
      attribute$length      lit  '4', /* no of bytes pr attribute */
      equal                  lit  '0', /* results of stringcompares */
      greater                lit  '1',
      less                   lit  '2';
```

```
/* Routine to compare tablestring with given symbolstring */
```

```
14 1  compare: proc byte;
15 2      i=0ffh;
16 2      do while (i:=i+1) < nochar;
17 3          if source(i) > look(i) then
18 3              return greater;
19 3          if source(i) < look(i) then
20 3              return less;
21 3          end$while;
22 2      return equal;
23 2  end compare;
```

```
/* Recursive routine to perform binary tablesearch */
```

```
24 1  binsearch: proc byte reent;
25 2      if noleft = 0 then$do
26 3          return false;
27 3      else$do
28 3          half=noleft/2;
29 3          t$lookahead=tok+half*nochar;
30 3          v$lookahead=value+half*attributelength;
31 3          do case compare;
32 3              /* equal */
33 3              do;
34 4                  value=v$lookahead; /* match found,pick up attributes */
35 5                  do i=0 to attributelength-1;
36 5                      dest(i)=valuebyte(i);
37 6                  end$do;
38 6              end$do;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____


```

39 5      return true;
40 5      end;

      /* greater */
41 4      do;
42 5      tok=t$lookahead+nochar; /* test last half of table */
43 5      value=v$lookahead+attributelen;
44 5      noleft=noleft-half-1;
45 5      return binsearch;
46 5      end;

      /* less */
47 4      do;
48 5      noleft=half; /* test first half of table */
49 5      return binsearch;
50 5      end;

51 4      end$case;
52 3      end$if;
53 2      end binsearch;

```

/*

Interface routine PFIND :

***** pfind (nochar,stringaddr,attributeaddr) byte *****

Routine to test if a given symbol is a predefined symbol.

entry: nochar = no of character in symbol
 stringaddr = address of symbol string
 attributeaddr = address to put the symbol-attributes (if found)

exit: The routine returns TRUE if symbol found, otherwise FALSE.

*/

```

54 1      pfind: proc (n,s,a) byte public;
55 2      dcl
          n      byte,
          (s,a)  address;

56 2      nochar=n; /* pick up parameters */
57 2      stringaddr=s;
58 2      attributeaddr=a;

59 2      if nochar < 10 then$do
61 3      value=valuepointer(nochar);
62 3      tok=tokpointer(nochar);
63 3      noleft=tokenbyte;
64 3      tok=tok+1;
65 3      return binsearch;
66 3      else$do
68 3      return false;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```
69 3      endif;
70 2      end pfind;

71 1      /***** end of module *****/
          end$module predef;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
54	0004H	2	A. WORD PARAMETER AUTOMATIC 55 58
2			ADDR LITERALLY 11
13	0002H	2	ATTRIBUTEADDR. WORD 13 37 53
13			ATTRIBUTELENGTH. LITERALLY 32 36 43
24	0049H	167	UNSEARCH. PROCEDURE BYTE REENTRANT STACK=0006H 45
4			BYT. LITERALLY 49 65
3			CODE LITERALLY 11
14	0000H	73	COMPARE. PROCEDURE BYTE STACK=0002H 33
2			CR LITERALLY
2			DCL. LITERALLY
3			DELETEDSYMB. LITERALLY
11	0001H	1	DESCR. BYTE MEMBER(VALUE9)
11	0001H	1	DESCR. BYTE MEMBER(VALUE4)
11	0001H	1	DESCR. BYTE MEMBER(VALUE6)
11	0001H	1	DESCR. BYTE MEMBER(VALUE2)
11	0001H	1	DESCR. BYTE MEMBER(VALUE3)
11	0001H	1	DESCR. BYTE MEMBER(VALUE7)
11	0001H	1	DESCR. BYTE MEMBER(VALUE5)
11	0001H	1	DESCR. BYTE MEMBER(VALUE8)
13	0000H	1	DEST. BYTE BASE(ATTRIBUTEADDR) ARRAY(1) 37
3			DOUBLEDEFINED. LITERALLY
4			DWRD LITERALLY 11
2			ELSEDD. LITERALLY
2			ENDCASE. LITERALLY
2			ENDDD. LITERALLY
2			ENDFOREVER. LITERALLY
2			ENDIF. LITERALLY
2			ENDMODULE. LITERALLY
2			ENDOFFILE. LITERALLY
2			ENDPROC. LITERALLY
2			ENDWHILE. LITERALLY 20
13			EQUAL. LITERALLY 22
3			ERROR. LITERALLY
2			FALSE. LITERALLY 27 68
2			FOREVER. LITERALLY
2			FORMFEED. LITERALLY
13			GREATER. LITERALLY 18
13	0010H	1	HALF. BYTE 30 31 32 44 48
13	0011H	1	I. BYTE 15 16 17 19 36 37
3			IDENT. LITERALLY
2			INIT. LITERALLY
3			LAB. LITERALLY
7			LEFTBRACKET. LITERALLY
13			LESS. LITERALLY 20
2			LF. LITERALLY
2			LIT. LITERALLY 2 3 4 5 6 7 8
13	0000H	1	LOOK. 13
			BYTE BASE(LOOKAHEAD) ARRAY(1) 17 19

CCP/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

54	000BH	1	N.	BYTE PARAMETER AUTOMATIC	55	56
8			NBYTE.	LITERALLY	11	
8			NWORD.	LITERALLY	11	
3			NEGLECTED.	LITERALLY		
4			NIL.	LITERALLY	11	
13	000EH	1	NOCCHAR.	BYTE	16	31 42 56 59 61 62
13	000FH	1	NOLEFT.	BYTE	25	30 44 48 63
3			NUMBER.	LITERALLY	11	
8			NWORD.	LITERALLY	11	
7			OAND.	LITERALLY	11	
7			OEQ.	LITERALLY	11	
7			OGE.	LITERALLY	11	
7			OGT.	LITERALLY	11	
7			OLAST.	LITERALLY	11	
7			OLE.	LITERALLY	11	
7			OLENGTH.	LITERALLY	11	
7			OLT.	LITERALLY	11	
7			OMOD.	LITERALLY	11	
7			ONE.	LITERALLY	11	
7			ONOT.	LITERALLY	11	
7			OFFSET.	LITERALLY	11	
7			ODR.	LITERALLY	11	
3			OPERATOR.	LITERALLY	11	
7			OPTR.	LITERALLY	11	
7			OSEG.	LITERALLY	11	
7			OSHL.	LITERALLY	11	
7			OSHORT.	LITERALLY	11	
7			OSHR.	LITERALLY	11	
7			OTYPE.	LITERALLY	11	
7			OXOR.	LITERALLY	11	
6			PCODEMACRO.	LITERALLY	11	
6			PCSEG.	LITERALLY	11	
6			PDB.	LITERALLY	11	
6			PDBTT.	LITERALLY	11	
6			PDD.	LITERALLY	11	
6			PDSEG.	LITERALLY	11	
6			PDW.	LITERALLY	11	
6			PEJECT.	LITERALLY	11	
6			PEND.	LITERALLY	11	
6			PENDIF.	LITERALLY	11	
6			PENOM.	LITERALLY	11	
6			PEQU.	LITERALLY	11	
6			PESEG.	LITERALLY	11	
54	00F0H	72	PFIND.	PROCEDURE BYTE PUBLIC STACK=0010h		
6			PIF.	LITERALLY	11	
6			PTFLIST.	LITERALLY	11	
6			PINCLUDE.	LITERALLY	11	
6			PLIST.	LITERALLY	11	
6			PMODRM.	LITERALLY	11	
6			PNOIFLIST.	LITERALLY	11	
6			PNOLIST.	LITERALLY	11	
6			PNOSEGFIX.	LITERALLY	11	
6			PORG.	LITERALLY	11	
6			PPAGESIZE.	LITERALLY	11	
6			PPAGewidth.	LITERALLY	11	
6			PRB.	LITERALLY	11	
1	0000H		PREDEF.	PROCEDURE STACK=0000h		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

6		PRELB.	.	.	.	.	.	LITERALLY	11				
6		PRELW.	.	.	.	.	.	LITERALLY	11				
2		PROG.	.	.	.	.	.	LITERALLY	14	24	54		
6		PRS.	.	.	.	.	.	LITERALLY	11				
6		PRW.	.	.	.	.	.	LITERALLY	11				
6		PSEGFTX.	.	.	.	.	.	LITERALLY	11				
3		PSEUDO.	.	.	.	.	.	LITERALLY	11				
6		PSIFORM.	.	.	.	.	.	LITERALLY	11				
6		PSSEG.	.	.	.	.	.	LITERALLY	11				
6		PTITLE.	.	.	.	.	.	LITERALLY	11				
5		RAH.	.	.	.	.	.	LITERALLY	11				
5		RAL.	.	.	.	.	.	LITERALLY	11				
5		RAX.	.	.	.	.	.	LITERALLY	11				
5		RBH.	.	.	.	.	.	LITERALLY	11				
5		RBL.	.	.	.	.	.	LITERALLY	11				
5		RBP.	.	.	.	.	.	LITERALLY	11				
5		RBX.	.	.	.	.	.	LITERALLY	11				
5		RCH.	.	.	.	.	.	LITERALLY	11				
5		RCL.	.	.	.	.	.	LITERALLY	11				
5		RCS.	.	.	.	.	.	LITERALLY	11				
5		RCX.	.	.	.	.	.	LITERALLY	11				
5		RDH.	.	.	.	.	.	LITERALLY	11				
5		RDI.	.	.	.	.	.	LITERALLY	11				
5		RDL.	.	.	.	.	.	LITERALLY	11				
5		RDS.	.	.	.	.	.	LITERALLY	11				
5		RDX.	.	.	.	.	.	LITERALLY	11				
2		REENT.	.	.	.	.	.	LITERALLY	24				
3		REG.	.	.	.	.	.	LITERALLY	11				
5		RES.	.	.	.	.	.	LITERALLY	11				
7		RIGHTBRACKET	.	.	.	.	.	LITERALLY					
5		RSI.	.	.	.	.	.	LITERALLY	11				
5		RSP.	.	.	.	.	.	LITERALLY	11				
5		RSS.	.	.	.	.	.	LITERALLY	11				
54	0006H	2	S.	.	.	.	.	WORD PARAMETER AUTOMATIC	55	57			
13	0000H	1	SOURCE	.	.	.	.	BYTE BASED(STRINGADDR) ARRAY(1)		17	19		
2			SPACE.	.	.	.	.	LITERALLY					
3			SPEC.	.	.	.	.	LITERALLY					
3			STRING	.	.	.	.	LITERALLY					
13	0000H	2	STRINGADDR	.	.	.	.	WORD	13	17	19	57	
2			STRUC.	.	.	.	.	LITERALLY	11				
3			SYMBOL	.	.	.	.	LITERALLY					
2			TAB.	.	.	.	.	LITERALLY					
2			THENDO	.	.	.	.	LITERALLY	25	59			
13	0008H	2	TLOOKAHEAD	.	.	.	.	WORD	13	17	19	31	42
13	0006H	2	TOK.	.	.	.	.	WORD	13	31	42	62	63 64
9	0158H	1	TOK0	.	.	.	.	BYTE ARRAY(1) DATA				10	
9	0159H	1	TOK1	.	.	.	.	BYTE ARRAY(1) DATA				10	
9	015AH	69	TOK2	.	.	.	.	BYTE ARRAY(69) DATA				10	
9	019FH	34	TOK3	.	.	.	.	BYTE ARRAY(34) DATA				10	
9	01C1H	53	TOK4	.	.	.	.	BYTE ARRAY(53) DATA				10	
9	01F6H	31	TOK5	.	.	.	.	BYTE ARRAY(31) DATA				10	
9	0215H	31	TOK6	.	.	.	.	BYTE ARRAY(31) DATA				10	
9	0234H	15	TOK7	.	.	.	.	BYTE ARRAY(15) DATA				10	
9	0243H	25	TOK8	.	.	.	.	BYTE ARRAY(25) DATA				10	
9	025CH	19	TOK9	.	.	.	.	BYTE ARRAY(19) DATA				10	
13	0000H	1	TOKENBYT.	.	.	.	.	BYTE BASED(TOK)	63				
10	0000H	20	TOKPOTINTER	.	.	.	.	WORD ARRAY(10) DATA				62	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		TRUE	LITERALLY	39	
11	0000H	1 TYPE	BYTE MEMBER(VALUE9)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE8)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE7)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE6)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE5)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE4)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE3)		
11	0000H	1 TYPE	BYTE MEMBER(VALUE2)		
3		UNDEFSYM	LITERALLY		
13	0004H	2 VALUE	WORD	13 32 35 37 43 51	
11	0002H	2 VALUE	WORD MEMBER(VALUE9)		
11	0002H	2 VALUE	WORD MEMBER(VALUE8)		
11	0002H	2 VALUE	WORD MEMBER(VALUE7)		
11	0002H	2 VALUE	WORD MEMBER(VALUE6)		
11	0002H	2 VALUE	WORD MEMBER(VALUE5)		
11	0002H	2 VALUE	WORD MEMBER(VALUE4)		
11	0002H	2 VALUE	WORD MEMBER(VALUE3)		
11	0002H	2 VALUE	WORD MEMBER(VALUE2)		
11	000CH	1 VALUE0	BYTE	12	
11	000DH	1 VALUE1	BYTE	12	
11	0014H	136 VALUE2	STRUCTURE ARRAY(34) DATA		12
11	009CH	44 VALUE3	STRUCTURE ARRAY(11) DATA		12
11	00C8H	52 VALUE4	STRUCTURE ARRAY(13) DATA		12
11	00FCH	24 VALUE5	STRUCTURE ARRAY(6) DATA		12
11	0114H	20 VALUE6	STRUCTURE ARRAY(5) DATA		12
11	0128H	8 VALUE7	STRUCTURE ARRAY(2) DATA		12
11	0130H	12 VALUE8	STRUCTURE ARRAY(3) DATA		12
11	013CH	8 VALUE9	STRUCTURE ARRAY(2) DATA		12
13	0000H	1 VALUEBYTE	BYTE BASED(VALUE) ARRAY(1)		37
12	0144H	20 VALUEPTR	WORD ARRAY(10) DATA		61
3		VARIABLE	LITERALLY		
13	000AH	2 VLOOKAHEAD	WORD	32 35 43	
4		WRD	LITERALLY	11	

MODULE INFORMATION:

CODE AREA SIZE = 0138H 3120
 CONSTANT AREA SIZE = 026FH 6230
 VARIABLE AREA SIZE = 0012H 180
 MAXIMUM STACK SIZE = 0010H 160
 462 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ASAS-11 PL/M-86 V2.0 COMPILATION OF MODULE ERRMSG

OBJECT MODULE PLACED IN ERMOD.OBJ

COMPILER INVOKED BY: :F0: ERMOD.PLM DEBUG OPTIMIZE(2) DATE(10/5/81) PAGELNGTH(100) XREF

1 \$title ("ERROR MESSAGE MODULE")

errorm:

do;

/*

modified 3/28/81 R. Silberstein

modified 3/30/81 R. Silberstein

modified 4/7/81 R. Silberstein

modified 4/24/81 R. Silberstein

*/

/*

This is the module to perform error message
printout to the print file. The interface from
other modules goes through the subroutine

ERRMSG (errornumber)

This routine also increments the global variable
"ERRORS" which contains the accumulated number
of errors throughout the assembly.

*/

```
= $include (:f1:macro.lit)
= $nolist
$include (:f1:struc.lit)
= $nolist
$include (:f1:ermod.lit)
= $nolist
$include (:f1:subr1.ext)
= $nolist
$include (:f1:subr2.ext)
= $nolist
$include (:f1:print.ext)
= $nolist
$include (:f1:global.ext)
= $nolist
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

/* Error messages : */

69 1

dcl

nulltext(1) byte data (0), /* dummy text */

tex00(*) byte data ("ILLEGAL FIRST ITEM",0),

tex01(*) byte data ("MISSING PSEUDO INSTRUCTION",0),

tex02(*) byte data ("ILLEGAL PSEUDO INSTRUCTION",0),

tex03(*) byte data ("DOUBLE DEFINED VARIABLE",0),

tex04(*) byte data ("DOUBLE DEFINED LABEL",0),

tex05(*) byte data ("UNDEFINED INSTRUCTION",0),

```

tex06(*) byte data ("GARBAGE AT END OF LINE - IGNORED",0),
tex07(*) byte data ("OPERAND(S) MISMATCH INSTRUCTION",0),
tex08(*) byte data ("ILLEGAL INSTRUCTION OPERANDS",0),
tex09(*) byte data ("MISSING INSTRUCTION",0),
tex10(*) byte data ("UNDEFINED ELEMENT OF EXPRESSION",0),
tex11(*) byte data ("ILLEGAL PSEUDO OPERAND",0),
tex12(*) byte data ("NESTED "IF" ILLEGAL - "IF" IGNORED",0),
tex13(*) byte data ("ILLEGAL "IF" OPERAND - "IF" IGNORED",0),
tex14(*) byte data ("NO MATCHING "IF" FOR "ENDIF",0),
tex15(*) byte data ("SYMBOL ILLEGALLY FORWARD REFERENCED - ",
    "NEGLECTED",0),
tex16(*) byte data ("DOUBLE DEFINED SYMBOL - ",
    "TREATED AS UNDEFINED",0),
tex17(*) byte data ("INSTRUCTION NOT IN CODE SEGMENT",0),
tex18(*) byte data ("FILE NAME SYNTAX ERROR",0),
tex19(*) byte data ("NESTED INCLUDE NOT ALLOWED",0),
tex20(*) byte data ("ILLEGAL EXPRESSION ELEMENT",0),
tex21(*) byte data ("MISSING TYPE INFORMATION IN OPERAND(S)",0),
tex22(*) byte data ("LABEL OUT OF RANGE",0),
tex23(*) byte data ("MISSING SEGMENT INFORMATION IN OPERAND",0),
tex24(*) byte data ("ERROR IN CODEMACROBUILDING",0),

```

```
/* Error-message pointer table: */
```

```

texttab(*) address data (.tex00,.tex01,.tex02,.tex03,.tex04,
    .tex05,.tex06,.tex07,.tex08,
    .tex09,.tex10,.tex11,.tex12,.tex13,
    .tex14,.tex15,.tex16,.tex17,.tex18,
    .tex19,.tex20,.tex21,.tex22,.tex23,
    .tex24,.nulltext);

```

```
/* Additional text strings: */
```

```

70 1 dcl
    errnotext(*) byte data ("** ERROR NO:",0),
    neartext(*) byte data ("** NEAR: ",0),
    spacetext(*) byte data (" ",0);

```

```
/* Table of defined error numbers: */
```

```

71 1 dcl
    errtab(*) byte data (firstitem,missingpseudo,
        illegalpseudo,doublededefvar,doublededeflab,
        illegalmacro,endsof$line$err,opmismatch,
        illioper,missinstr,undefsymbol,
        pseudooperr,nestediferr,ifparerr,
        missiferr,neglecterr,doubledefsymp,
        instrerr,filesynterr,
        nestedincludeerr,illexprelem,misstypeinfo,
        laboutofrange,misssegminfo,codemacroerr);

```

```
/* Subroutines: */
```

```

72 1 printtext: proc(txt);
73 2   dcl txt address,ch based txt (1) byte,i byte;
74 2   i=0ffh;
75 2   do while ch(i:=i+1) <> 0;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

76 3      call printsinglebyte(ca(i));
77 3      end$while;
78 2      end printtext;

79 1      locerrmsg: proc(erno);
80 2      dcl t address,help(5) byte,(helpstop,erno,i) byte;

81 2      errortype: proc byte;
82 3      i=0ffh;
83 3      do while (i:=i+1) < length(errtab);
84 4          if erno = errtab(i) then return i;
            end$while;
87 3      return length(errtab);
88 3      end errortype;

89 2      helpstop,accum(acclen)=0;
90 2      call decout(erno,.help(0));

91 2      t=texttab(errortype); /* pick up correct error text */

92 2      call printtext(.errnotext); /* print error message line */
93 2      call printtext(.help(2));
94 2      if accum(0) <> cr then$do
95 3          call printtext(.neartext);
96 3          call printtext(.accum(0)); /*.(print current token) */
97 3          CALL PRINTSINGLEBYTE ("*");
98 3          end$if;
99 3          call printtext(.spacetext);
100 2          call printtext(t);
101 2          call printerlf;
102 2          end locerrmsg;

/*      Public routine:      */

104 1      errmsg: proc(erno) public;
105 2      dcl erno byte;
106 2      if print$on OR PRINTSWITCHOFF then$do
107 3          if not errorprinted then$do
108 4              errorprinted=true;
109 4              call locerrmsg(erno);
110 4              errors=errors+1;
111 4              end$if;
112 4              end$if;
113 4              end errmsg;
114 2          end errmsg;
115 2          end errmsg;

116 1      end$module errorm;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
66	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(54)
66	0000H	1	ACLEN BYTE EXTERNAL(58) 89
66	0000H	1	ACCLNSAVE BYTE EXTERNAL(61)
66	0000H	30	ACCUM. BYTE ARRAY(30) EXTERNAL(59) 39 94 97
66	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(60)
2			ADDR LITERALLY 6 9 12 15 21 39 42
			45 48 66
29	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(8) STACK=0000H
32	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(9) STACK=0000H
66	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
66	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
3			BASEREGBIT LITERALLY
3			BASEREGCOUNT LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
23	0000H	1	CH BYTE PARAMETER 24
32	0000H	1	CH BYTE PARAMETER 33
26	0000H	1	CH BYTE PARAMETER 27
35	0000H	1	CH BYTE PARAMETER 36
20	0000H	1	CH BYTE PARAMETER 21
73	0000H	1	CH BYTE BASED(TXT) ARRAY(1) 75 76
50	0000H	1	CH BYTE PARAMETER 51
29	0000H	1	CH BYTE PARAMETER 30
66	0000H	2	CIP. WORD EXTERNAL(23)
4			CODEMACROERR LITERALLY 71
66	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(80)
3			CODEMACROHEAD. LITERALLY
66	0000H	2	CODEMACROPTR WORD EXTERNAL(75)
11	0000H		COPY PROCEDURE EXTERNAL(2) STACK=0000H
2			CR LITERALLY 94
66	0000H	2	CSCIP. WORD EXTERNAL(28)
66	0000H	1	CSEGSPEC BYTE EXTERNAL(26)
66	0000H	1	CSECTYPE BYTE EXTERNAL(24)
66	0000H	2	CSEGVALUE. WORD EXTERNAL(25)
66	0000H	1	CSPEC. BYTE EXTERNAL(35)
66	0000H	2	CURCSEG. WORD EXTERNAL(31)
66	0000H	2	CURDSEG. WORD EXTERNAL(32)
66	0000H	2	CURESEG. WORD EXTERNAL(34)
66	0000H	9	CURRENTSYMBOL. STRUCTURE EXTERNAL(71)
66	0000H	2	CURSSEG. WORD EXTERNAL(33)
47	0000H	2	D. WORD PARAMETER 48
38	0000H	2	D. WORD PARAMETER 39
41	0000H	2	D. WORD PARAMETER 42
11	0000H	2	D. WORD PARAMETER 12
44	0000H	2	D. WORD PARAMETER 45
2			DCL. LITERALLY
47	0000H		DECOUT PROCEDURE EXTERNAL(14) STACK=0000H 90
66	0000H	1	DEFAULTDRIVE BYTE EXTERNAL(78)
66	0001H	1	DESCR. BYTE MEMBER(TOKEN)

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

23	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(6) STACK=0000H
4			DOUBLEDEFVAR . . .	LITERALLY 71
4			DOUBLEDEFSYMB. . .	LITERALLY 71
4			DOUBLEDEFVAR . . .	LITERALLY 71
66	0000H	2	DSCVP.	WORD EXTERNAL(30)
66	0000H	1	DSPEC.	BYTE EXTERNAL(36)
64	0000H		EJECT.	PROCEDURE EXTERNAL(21) STACK=0000H
2			ELSEDF.	LITERALLY
2			ENDCASE.	LITERALLY
2			ENDDD.	LITERALLY
2			ENDFOREVER. . . .	LITERALLY
2			ENDIF.	LITERALLY 113
2			ENDMODULE.	LITERALLY
2			ENDOFFILE.	LITERALLY
4			ENDOFFLINEERR. . .	LITERALLY 71
2			ENDPROC.	LITERALLY
2			ENDWHILE.	LITERALLY 85
66	0000H	1	EOFSET.	BYTE EXTERNAL(62)
38	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(11) STACK=0000H
79	0000H	1	ERNO.	BYTE PARAMETER 80 84 90
104	0004H	1	ERNO.	BYTE PARAMETER AUTOMATIC 105 111
104	00C1H	42	ERRMSG.	PROCEDURE PUBLIC STACK=0012H
70	0314H	13	ERRNOTEXT.	BYTE ARRAY(13) DATA 92
1	0000H		ERRORM.	PROCEDURE STACK=0000H
66	0000H	1	ERRORPRINTED. . .	BYTE EXTERNAL(65) 108 110
66	0000H	2	ERRORS.	WORD EXTERNAL(55) 112
81	0099H	40	ERRORTYPE.	PROCEDURE BYTE STACK=0002H 91
71	0332H	25	ERRTAB.	BYTE ARRAY(25) DATA 83 84 87
66	0000H	2	ESCP.	WORD EXTERNAL(27)
66	0000H	1	ESPEC.	BYTE EXTERNAL(38)
2			FALSE.	LITERALLY
17	0000H	2	FCBADR.	WORD PARAMETER 18
17	0000H		FILEABORT.	PROCEDURE EXTERNAL(4) STACK=0000H
3			FILEISTRUCTURE. . .	LITERALLY
3			FILEOSTRUCTURE. . .	LITERALLY
4			FILESYNTER.	LITERALLY 71
20	0000H		FILL.	PROCEDURE EXTERNAL(5) STACK=0000H
4			FIRSTITEM.	LITERALLY 71
66	0000H	2	FIRSTMACROPTR. . .	WORD EXTERNAL(66)
2			FOREVER.	LITERALLY
2			FORMFELD.	LITERALLY
66	0000H	1	FULLSYMBTAB. . . .	BYTE EXTERNAL(68)
67	0000H		GLOBALINIT.	PROCEDURE EXTERNAL(81) STACK=0000H
66	0000H	5	HELP.	BYTE ARRAY(5) EXTERNAL(76)
80	0005H	5	HELP.	BYTE ARRAY(5) 90 93
80	000AH	1	HELPSTOP.	BYTE 89
41	0000H		HEX1OUT.	PROCEDURE EXTERNAL(12) STACK=0000H
44	0000H		HEX2OUT.	PROCEDURE EXTERNAL(13) STACK=0000H
66	0000H	1	I.	BYTE EXTERNAL(77)
80	0006H	1	I.	BYTE 82 83 84 85
73	0004H	1	I.	BYTE 74 75 76
66	0000H	1	IFLEVEL.	BYTE EXTERNAL(70)
66	0000H	1	IFLIST.	BYTE EXTERNAL(41)
4			IFPARERR.	LITERALLY 71
4			ILLEGALMACRO. . . .	LITERALLY 71
4			ILLEGALPSEUDO. . .	LITERALLY 71
4			ILLEXPRELEM. . . .	LITERALLY 71

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

4		ILLIOPER	LITERALLY	71
66	0000H	1 INCLUDEDDEFAULT . .	BYTE EXTERNAL(79)	
66	0000H	1 INCLUDEON	BYTE EXTERNAL(59)	
3		INDEXREGBIT	LITERALLY	
3		INDEXREGCOUNT . . .	LITERALLY	
2		INIT	LITERALLY	
4		INSTERR	LITERALLY	71
66	0000H	1 INTELHEXON	BYTE EXTERNAL(63)	
3		IREGBIT	LITERALLY	
3		IREGCOUNT	LITERALLY	
4		LABOUTOFKANGLE . . .	LITERALLY	71
66	0000H	2 LENGTH	WORD MEMBER(OPERANDS)	
66	0000H	2 LENGTH	WORD MEMBER(CURRENTSYMBOL)	
		LENGTH	BUILTIN	83 87
26	0000H	LETTER	PROCEDURE BYTE EXTERNAL(7) STACK=0000H	
2		LF	LITERALLY	
2		LIT	LITERALLY	2 3 4
79	0027H	114 LOCERRMSG	PROCEDURE STACK=0000H	111
66	0000H	2 MACROPTR	WORD EXTERNAL(67)	
66	0000H	1 MAXCOL	BYTE EXTERNAL(42)	
4		MISSIFERR	LITERALLY	71
4		MISSINGPSEUDO	LITERALLY	71
4		MISSINSTR	LITERALLY	71
4		MISSEGMINFO	LITERALLY	71
4		MISSTYPEINFO	LITERALLY	71
47	0000H	2 N	WORD PARAMETER	43
44	0000H	2 N	WORD PARAMETER	45
41	0000H	1 N	BYTE PARAMETER	42
38	0000H	1 N	BYTE PARAMETER	39
20	0000H	1 N	BYTE PARAMETER	21
11	0000H	1 N	BYTE PARAMETER	12
8	0000H	2 N	WORD PARAMETER	9
70	0321H	13 NEARTEXT	BYTE ARRAY(13) DATA	96
4		NEGLECTERR	LITERALLY	71
4		NESTEDIFERR	LITERALLY	71
4		NESTEDINCLUDEERR . .	LITERALLY	71
66	0000H	1 NEXTCH	BYTE EXTERNAL(57)	
66	0000H	1 NOERRR	BYTE EXTERNAL(64)	
66	0000H	1 NOOPER	BYTE EXTERNAL(73)	
3		NOOVERCOUNT	LITERALLY	
3		NOOVERRIDEBIT	LITERALLY	
69	0034H	1 NULLTEXT	BYTE ARRAY(1) DATA	69
66	0006H	2 OFFSET	WORD MEMBER(OPERANDS)	
66	0006H	2 OFFSET	WORD MEMBER(CURRENTSYMBOL)	
66	0000H	36 OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(74)	
3		OPERANDSTRUC	LITERALLY	66
4		OPMTSMATCH	LITERALLY	71
14	0000H	OUTTEXT	PROCEDURE EXTERNAL(3) STACK=0000H	
66	0000H	1 PAGESIZE	BYTE EXTERNAL(48)	
66	0000H	1 PASS	BYTE EXTERNAL(22)	
66	0000H	240 PREFIX	BYTE ARRAY(240) EXTERNAL(52)	
66	0000H	1 PREFIXPTR	BYTE EXTERNAL(53)	
53	0000H	PRINTCRLF	PROCEDURE EXTERNAL(16) STACK=0000H	102
66	0000H	1 PRINTDEVICE	BYTE EXTERNAL(45)	
59	0000H	PRINTINIT	PROCEDURE EXTERNAL(19) STACK=0000H	
55	0000H	PRINTNEWPAGE	PROCEDURE EXTERNAL(17) STACK=0000H	
66	0000H	1 PRINTON	BYTE EXTERNAL(39)	106

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

50	0000H		PRINTSINGLEBYTE. .	PROCEDURE EXTERNAL(15) STACK=0000H	76	93
57	0000H		PRINTSOURCELINE. .	PROCEDURE EXTERNAL(16) STACK=0000H		
66	0000H	1	PRINTSWITCHOFF. .	BYTE EXTERNAL(49) 106		
61	0000H		PRINTTERMINATE. .	PROCEDURE EXTERNAL(20) STACK=0000H		
72	0000H	39	PRINTTEXT.	PROCEDURE STACK=0000H 92	95	96 97
				100 101		
2			PROC.	LITERALLY 5 8 29 32 41 44 47		
				50 53 55 57 59 61 64 72 79 81		
				104		
4			PSEUDOOPERR. . . .	LITERALLY 71		
20	0000H	2	PT.	WORD PARAMETER 21		
2			REENT.	LITERALLY		
38	0000H	2	S.	WORD PARAMETER 59		
11	0000H	2	S.	WORD PARAMETER 12		
66	0000H	12	SAVESOURCE. . . .	BYTE ARRAY(12) EXTERNAL(44)		
3			SEGMBIT.	LITERALLY		
3			SEGMCOUNT.	LITERALLY		
66	0004H	2	SEGMENT.	WORD MEMBER(OPERANDS)		
66	0004H	2	SEGMENT.	WORD MEMBER(CURRENTSYMBOL)		
3			SEGTYPEBIT. . . .	LITERALLY		
3			SEGTYPECOUNT. .	LITERALLY		
66	0003H	1	SFLAG.	BYTE MEMBER(OPERANDS)		
66	0003H	1	SFLAG.	BYTE MEMBER(CURRENTSYMBOL)		
66	0000H	1	SIMFORM.	BYTE EXTERNAL(49)		
66	0000H	80	SOURCEBUF.	BYTE ARRAY(80) EXTERNAL(50)		
66	0000H	12	SOURCENAME. . . .	BYTE ARRAY(12) EXTERNAL(43)		
66	0000H	1	SOURCEPTR.	BYTE EXTERNAL(51)		
2			SPACE.	LITERALLY		
70	032EH	4	SPACETEXT.	BYTE ARRAY(4) DATA 100		
66	0000H	2	SSCTP.	WORD EXTERNAL(29)		
66	0000H	1	SSPEC.	BYTE EXTERNAL(37)		
2			STRUC.	LITERALLY 66		
66	0002H	1	STYPE.	BYTE MEMBER(OPERANDS)		
66	0002H	1	STYPE.	BYTE MEMBER(CURRENTSYMBOL)		
66	0000H	1	SYMBOLDEVICE. . .	BYTE EXTERNAL(46)		
3			SYMBOLHEAD. . . .	LITERALLY		
3			SYMBOLSTRUC. . . .	LITERALLY 66		
66	0000H	2	SYMBTABADR. . . .	WORD EXTERNAL(72)		
80	0002H	2	T.	WORD 91 101		
14	0000H	2	T.	WORD PARAMETER 15		
2			TAB.	LITERALLY		
69	0035H	19	TEX00.	BYTE ARRAY(19) DATA 69		
69	0048H	27	TEX01.	BYTE ARRAY(27) DATA 69		
69	0063H	27	TEX02.	BYTE ARRAY(27) DATA 69		
69	007EH	24	TEX03.	BYTE ARRAY(24) DATA 69		
69	0096H	21	TEX04.	BYTE ARRAY(21) DATA 69		
69	00ABH	22	TEX05.	BYTE ARRAY(22) DATA 69		
69	00C1H	33	TEX06.	BYTE ARRAY(33) DATA 69		
69	00E2H	37	TEX07.	BYTE ARRAY(32) DATA 69		
69	0102H	29	TEX08.	BYTE ARRAY(29) DATA 69		
69	011FH	20	TEX09.	BYTE ARRAY(20) DATA 69		
69	0133H	32	TEX10.	BYTE ARRAY(32) DATA 69		
69	0155H	23	TEX11.	BYTE ARRAY(23) DATA 69		
69	016AH	35	TEX12.	BYTE ARRAY(35) DATA 69		
69	018DH	36	TEX13.	BYTE ARRAY(36) DATA 69		
69	01B1H	29	TEX14.	BYTE ARRAY(29) DATA 69		
69	01CEH	48	TEX15.	BYTE ARRAY(48) DATA 69		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

69	01FEH	45	TEX16.	BYTE ARRAY(45) DATA	69
69	022BH	32	TEX17.	BYTE ARRAY(32) DATA	69
69	024BH	23	TEX18.	BYTE ARRAY(23) DATA	69
69	0262H	27	TEX19.	BYTE ARRAY(27) DATA	69
69	027DH	27	TEX20.	BYTE ARRAY(27) DATA	69
69	0298H	39	TEX21.	BYTE ARRAY(39) DATA	69
69	02BFH	19	TEX22.	BYTE ARRAY(19) DATA	69
69	02D2H	39	TEX23.	BYTE ARRAY(39) DATA	69
69	02F9H	27	TEX24.	BYTE ARRAY(27) DATA	69
17	0000H	2	TEXTADR.	WORD PARAMETER	18
69	0000H	52	TEXTTAB.	WORD ARRAY(26) DATA	91
2			THENDO.	LITERALLY	94 106 108
66	0000H	30	TITLE.	BYTE ARRAY(30) EXTERNAL(47)	
66	0000H	4	TOKEN.	STRUCTURE EXTERNAL(56)	
2			TRUE.	LITERALLY	110
72	0004H	2	TXT.	WORD PARAMETER AUTOMATIC	73 75 76
66	0000H	1	TYPE.	BYTE MEMBER(TOKEN)	
3			TYPEBIT.	LITERALLY	
5	0000H		TYPECALC.	PROCEDURE BYTE EXTERNAL(0) STACK=0000H	
3			TYPECOUNT.	LITERALLY	
4			UDFSYMBOL.	LITERALLY	71
35	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(10) STACK=0000H	
61	0000H	1	USEFACT.	BYTE PARAMETER	62
5	0000H	2	VAL.	WORD PARAMETER	6
66	0002H	2	VALUE.	WORD MEMBER(TOKEN)	
8	0000H		WRDTEST.	PROCEDURE BYTE EXTERNAL(1) STACK=0000H	

MODULE INFORMATION:

```

CODE AREA SIZE      = 002BH      2350
CONSTANT AREA SIZE  = 034BH      8430
VARIABLE AREA SIZE  = 000CH      120
MAXIMUM STACK SIZE  = 0012H      180
523 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE TEXT

OBJECT MODULE PLACED IN TEXT.OBJ

COMPILER INVOKED BY: :FO: TEXT.PLA DEBUG OPTIMIZE(2) DATE(10/5/81) PAGESIZE(100) XLAT

```

1      $title ("TEXT MODULE")
      text:
      do:
      $include (:f1:macro.lit)
      $nolist

      /*
      modified 5/28/81 R. Silberstein
      modified 3/30/81 R. Silberstein
      modified 4/8/81 R. Silberstein
      modified 4/15/81 R. Silberstein
      modified 7/24/81 R. Silberstein

      */
      /* Bytes to define "yes" and "no": */

```

```

3 1    dcl yes byte public data ('Y'),
      no byte public data ('N');

```

```

      /* Text strings: */

4 1    dcl
      initials(*) byte public data("CP/M ASM86 1.1 SOURCE: ",0),
      pagetext(*) byte public data(" PAGE",0),
      endtext(*) byte public data("END OF ASSEMBLY. ",
      "NUMBER OF ERRORS: ",0),
      USEFACTOR(*) BYTE PUBLIC DATA (" USE FACTOR: ",0),
      as86text(*) byte public data(cr,lf,"CP/M 8086 ASSEMBLER VER 1.1",
      cr,lf,0),
      paramerrtext(*) byte public data(cr,lf,"PARAMETER ERROR",cr,lf,0),
      openerrtext(*) byte public data(cr,lf,"NO FILE",0),
      DISKREADERRTEXT (*) BYTE PUBLIC DATA (CR,LF,"DISK READ ERROR",0),
      DISKWRITEERRTEXT (*) BYTE PUBLIC DATA (CR,LF,"DISK FULL",0),
      MAKEERRTEXT (*) BYTE PUBLIC DATA (CR,LF,"DIRECTORY FULL",0),
      CLOSEERRTEXT (*) BYTE PUBLIC DATA (CR,LF,"CANNOT CLOSE",0),
      SYMBFULLERRTEXT (*) BYTE PUBLIC DATA (CR,LF,"SYMBOL TABLE OVERFLOW",0),
      usbreaktext(*) byte public data(cr," USER BREAK. OK(Y/N)? ",0),
      pass0text (*) byte public data("END OF PASS 1",cr,lf,0),
      pass1text (*) byte public data("END OF PASS 2",cr,lf,0);

```

```

5 1    end$module text;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
2			ADDR LITERALLY
4	0056H	32	ASH86TEXT. BYTE ARRAY(32) PUBLIC DATA
4	00C3H	15	CLUSEERRTEXT BYTE ARRAY(15) PUBLIC DATA
2			CR LITERALLY 4
2			DCL LITERALLY
4	0094H	18	DISKREADERRTEXT. BYTE ARRAY(18) PUBLIC DATA
4	00A6H	12	DISKWRITEERRTEXT. BYTE ARRAY(12) PUBLIC DATA
2			ELSEDD LITERALLY
2			ENDCASE. LITERALLY
2			ENDDO. LITERALLY
2			ENDFOREVER LITERALLY
2			ENDIF. LITERALLY
2			ENDMODULE. LITERALLY
2			ENDOFFILE. LITERALLY
2			ENDPROC. LITERALLY
4	0021H	37	ENDTEXT. BYTE ARRAY(37) PUBLIC DATA
2			ENDWHILE LITERALLY
2			FALSE. LITERALLY
2			FOREVER. LITERALLY
2			FORMFEED LITERALLY
2			INIT LITERALLY
4	0002H	25	INITIALS BYTE ARRAY(25) PUBLIC DATA
2			LF LITERALLY 4
2			LIT. LITERALLY 2
4	00B2H	17	MAKEERRTEXT. BYTE ARRAY(17) PUBLIC DATA
3	0001H	1	NO BYTE PUBLIC DATA
4	008AH	10	OPENERTEXT. BYTE ARRAY(10) PUBLIC DATA
4	001BH	6	PAGETEXT BYTE ARRAY(6) PUBLIC DATA
4	0076H	20	PARERRTEXT BYTE ARRAY(20) PUBLIC DATA
4	0102H	16	PASSTEXT. BYTE ARRAY(16) PUBLIC DATA
4	0112H	16	PASSITEXT. BYTE ARRAY(16) PUBLIC DATA
2			PROC LITERALLY
2			REENT. LITERALLY
2			SPACE. LITERALLY
2			STRUC. LITERALLY
4	00D2H	24	SYMBFULLERRTEXT. BYTE ARRAY(24) PUBLIC DATA
2			TAB. LITERALLY
1	0000H		TEXT PROCEDURE STACK=0000H
2			THENDD LITERALLY
2			TRUE LITERALLY
4	00EAH	24	USBREAKEXT. BYTE ARRAY(24) PUBLIC DATA
4	0046H	16	USEFACTOR. BYTE ARRAY(16) PUBLIC DATA
3	0000H	1	YES. BYTE PUBLIC DATA

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

MODULE INFORMATION:

CODE AREA SIZE = 0000H 00

PL/M-86 COMPILER TEXT MODULE

10/2/83 10.00 2

CONSTANT AREA SIZE = 0122H 2900
VARIABLE AREA SIZE = 0000H 00
MAXIMUM STACK SIZE = 0000H 00
77 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE HEXOUT
 OBJECT MODULE PLACED IN OUTP.OBJ
 COMPILER INVOKED BY: :F0: OUTP.PLX DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) YREF

```
1 $title ("HEX OUTPUT MODULE")
```

```
hexout:
```

```
do;
```

```
/*
```

```
modified 3/28/81 R. Silberstein
```

```
modified 3/30/81 R. Silberstein
```

```
modified 4/9/81 R. Silberstein
```

```
*/
```

```
/*
```

This is the module to produce the (hex-)output
 from the assembler. The interface to other modules
 goes through the subroutine

```
EMITCODEBYTE (outputbyte,segmenttype).
```

This routine outputs one byte of generated code of
 a specified segment type (code,data,stack,extra).

The subroutine also updates the value of the current
 instruction pointer of the current segment (CIP),
 and prints the output code on the print line.

```
*/
```

```
= $include (:f1:macro.lit)
= $nolist
= $include (:f1:struc.lit)
= $nolist
= $include (:f1:outp.lit)
= $nolist
= $include (:f1:subr2.ext)
= $nolist
= $include (:f1:files.ext)
= $nolist
= $include (:f1:global.ext)
= $nolist
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
84 1 dcl
empty lit "0ffh", /* buffer empty value */
recordlimit lit "30", /* max no of bytes pr record */
loccip addr, /* local copy of instruction pointer */
startfound byte, /* true if start record sent */
gtyp byte, /* incoming byte type */
gbyt byte, /* incoming byte */
curtyp byte, /* current byte type */
sum byte, /* used to compute check sum */
buffer (35) byte, /* record buffer (RECORDLIMIT+5) */
```

```

recordlg    byte at (.buffer(0)),
recordtype  byte at (.buffer(3)),
offsetaddr  addr at (.buffer(1)),
bufpt      byte,      /* buffer index */

```

```

/* Record type conversion table */
/* ( to be changed later ??? ) */

```

```
rectyp$I$tab(12) byte data
```

```

(Offh,eoftype,Offh,starttype,INTELdata,INTELdata,
 INTELdata,INTELdata,INTELsegment,INTELsegment,INTELsegment,
 INTELsegment),

```

```
rectyp$D$tab(12) byte data
```

```

(Offh,eoftype,Offh,starttype,DRcodedata,DRdatadata,
 DRstackdata,DRextradata,DRcodesegm,DRdatasegm,DRstacksegm,
 DRextrasegm);

```

```
/****** subroutines *****/
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

85 1 rectyptab: procedure(n) byte;
86 2   declare n byte;
87 2   if intel$hex$on then$do
89 3     return rectyp$I$tab(n);
90 3   else$do
92 3     return rectyp$D$tab(n);
93 3   end$if;
94 2 end rectyptab;

95 1 switch$high$low: procedure(p);
96 2   declare p address, ch based p byte, (s1,s2) byte;
97 2   s1=ch;
98 2   p=p+1;
99 2   s2=ch;
100 2   ch=s1;
101 2   p=p-1;
102 2   ch=s2;
103 2 end switch$high$low;

104 1 writebyt: proc (ch);
105 2   dcl ch byte;
106 2   call outhexbyte(ch);
107 2 end writebyt;

108 1 writerecord: proc; /* write current recor to file */
109 2   call switch$high$low(.offsetaddr);
110 2   recordlg=bufpt-4;
111 2   sum=0; /* compute check sum */
112 2   i=0ffh;
113 2   do while (i:=i+1) < bufpt;
114 3     sum=sum+buffer(i);
115 3   end$while;
116 2   buffer(bufpt)=-sum; /* check sum */
117 2   call writebyt(":");

```

```

118 2      do i=0 to bufpt;      /* print hexbytes to file */
119 3          call hexlout(buffer(i),.help(0));
120 3          call writebyt(.help(0));
121 3          call writebyt(.help(1));
122 3      enddo;
123 2      call writebyt(cr);
124 2      call writebyt(lf);
125 2      end writerecord;

126 1      enternewbyt: proc(b); /* enter a new byte into buffer */
127 2          dcl b byte;
128 2          if bufpt > recordlimit then$do /* test if record full */
129 3              call writerecord;
130 3              offsetaddr=cip;
131 3              bufpt=4;
132 3          end$if;
133 2          buffer(bufpt)=b;
134 2          bufpt=bufpt+1;
135 2      end enternewbyt;

137 1      enterinput: proc;
138 2          call enternewbyt(qbyt);
139 2      end enterinput;

140 1      eofrecord: proc; /* write end-of-file record to file */
141 2          if curtyp<>empty then call writerecord;
142 2          recordtype=rectyptab(eoftype);
143 2          offsetaddr=0;
144 2          bufpt=4;
145 2          call writerecord;
146 2      end eofrecord;

148 1      startrecord: proc; /* write a start record to file */
149 2          dcl seglow byte at (.csegvalue),seghigh byte at (.csegvalue+1),
150 2              offslow byte at (.cip),offshigh byte at (.cip+1);
151 2          if pass=2 then$do
152 3              startfound=true;
153 3              if curtyp <> empty then call writerecord;
154 3              bufpt=4;
155 3              offsetaddr=0;
156 3              recordtype=rectyptab(starttype);
157 3              if csegspec then$do
158 4                  call enternewbyt(seghigh);
159 4                  call enternewbyt(seglow);
160 4              else$do
161 4                  call enternewbyt(0);
162 4                  call enternewbyt(0);
163 4              end$if;
164 3              call enternewbyt(offshigh);
165 3              call enternewbyt(offslow);
166 3              call writerecord;
167 3              curtyp=empty;
168 3          end$if;
169 2      end startrecord;

173 1      segmbyte: proc; /* write a segment value byte to file */
174 2          if pass = 2 then$do

```

CCP/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

```
176 3      if curtyp <> gtyp then$do
178 4          if curtyp <> empty then call writerecord;
180 4          curtyp=gtyp;
181 4          recordtype=rectypstab(gtyp);
182 4          offsetaddr=0;
183 4          bufpt=4;
184 4          call enterinput;
185 4      else$do
187 4          call enterinput;
188 4          call writerecord;
189 4          curtyp=empty;
190 4      end$if;
191 3      end$if;
192 2      end segmbyte;

193 1      databyte: proc; /* write a data byte to file */
194 2          if pass=2 then$do
196 3              if (curtyp <> gtyp) or (loccip <> cip) then$do
198 4                  if curtyp<>empty then call writerecord;
200 4                  curtyp=gtyp;
201 4                  recordtype=rectypstab(gtyp);
202 4                  offsetaddr=cip;
203 4                  bufpt=4;
204 4                  end$if;
205 3                  call enterinput;
206 3                  call hexlout(gbyt,.prefix(prefixptr)); /* output to listing */
207 3                  prefixptr=prefixptr+2;
208 3                  end$if;
209 2                  cip=cip+1; /* update instruction pointer */
210 2                  loccip=cip;
211 2              end databyte;

212 1      emitinit: proc public;
213 2          startfound=false;
214 2          curtyp=empty;
215 2      end emitinit;

216 1      emitterminate: proc public;
217 2          call eofrecord; /* write EOF record */
218 2      end emitterminate;

219 1      emitcodebyte: proc (b,typ) public;
220 2          dcl (b,typ) byte;
221 2          gbyt=b; /* move to global variables */
222 2          gtyp=typ;

223 2          do case typ-CSdata;

224 3              do; /* CS data */
225 4                  if not startfound then$do
227 5                      call startrecord;
228 5                  end$if;
229 4                  call databyte;
230 4              end;
231 3              call databyte; /* DS data */
232 3              call databyte; /* SS data */
233 3              call databyte; /* ES data */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
234 3      call segmbyte; /* CS value */
235 3      call segmbyte; /* DS value */
236 3      call segmbyte; /* SS value */
237 3      call segmbyte; /* ES value */

238 3      end$case;

239 2      end emitcodebyte;

240 1      end$module hexout;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
81	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(62)
81	0000H	1	ACCLN. BYTE EXTERNAL(65)
81	0000H	1	ACCLNSAVE. BYTE EXTERNAL(69)
81	0000H	80	ACCUH. BYTE ARRAY(80) EXTERNAL(67)
81	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(68)
2			ADDR. LITERALLY 6 12 30 33 36 39 77 81 84
20	0000H		ALPHANUMERIC. PROCEDURE BYTE EXTERNAL(5) STACK=0000H
23	0000H		ASCTCHAR. PROCEDURE BYTE EXTERNAL(6) STACK=0000H
219	0006H	1	B. BYTE PARAMETER AUTOMATIC 220 221
126	0004H	1	B. BYTE PARAMETER AUTOMATIC 127 134
81	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
81	0008H	1	BASEINDEX. BYTE MEMBER(COOPERANDS)
3			BASEREGBIT. LITERALLY
3			BASEREGCOUNT. LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
84	0007H	35	BUFFER. BYTE ARRAY(35) 84 114 116 119 134
84	002AH	1	SUFPT. BYTE 110 113 116 118 128 132 134 135 145 155 183 203
14	0000H	1	CH. BYTE PARAMETER 15
104	0004H	1	CH. BYTE PARAMETER AUTOMATIC 105 106
47	0000H	1	CH. BYTE PARAMETER 48
41	0000H	1	CH. BYTE PARAMETER 42
26	0000H	1	CH. BYTE PARAMETER 27
20	0000H	1	CH. BYTE PARAMETER 21
11	0000H	1	CH. BYTE PARAMETER 12
23	0000H	1	CH. BYTE PARAMETER 24
44	0000H	1	CH. BYTE PARAMETER 45
96	0000H	1	CH. BYTE BASED(P) 97 99 100 102
17	0000H	1	CH. BYTE PARAMETER 18
81	0000H	2	CIP. WORD EXTERNAL(31) 131 149 196 202 209 210
70	0000H		CLOSEHEX. PROCEDURE EXTERNAL(25) STACK=0000H
68	0000H		CLOSEINCLUDE. PROCEDURE EXTERNAL(24) STACK=0000H
72	0000H		CLOSEPTNT. PROCEDURE EXTERNAL(26) STACK=0000H
64	0000H		CLOSESOURCE. PROCEDURE EXTERNAL(22) STACK=0000H
74	0000H		CLOSESYMBOL. PROCEDURE EXTERNAL(27) STACK=0000H
81	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(89)
3			CODEMACROHEAD. LITERALLY
81	0000H	2	CODEMACROPTR. WORD EXTERNAL(83)
2			CR. LITERALLY 123
81	0000H	2	CSCTP. WORD EXTERNAL(36)
4			CSDATA. LITERALLY 223
81	0000H	1	CSEGSPLC. BYTE EXTERNAL(34) 158
81	0000H	1	CSECTYPE. BYTE EXTERNAL(32)
81	0000H	2	CSEGVVALUE. WORD EXTERNAL(33) 149
81	0000H	1	CSPEC. BYTE EXTERNAL(43)
4			CSVALUE. LITERALLY

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

81	0000H	2	CURCSEG.	WORD EXTERNAL(39)	
81	0000H	2	CURDSEG.	WORD EXTERNAL(40)	
81	0000H	2	CURESEG.	WORD EXTERNAL(42)	
81	0000H	9	CURRENTSYMBOL.	STRUCTURE EXTERNAL(79)	
81	0000H	2	CURSSEG.	WORD EXTERNAL(41)	
84	0005H	1	CURTYP.	BYTE	141 153 170 176 178 180 182 186
					198 200 214
38	0000H	2	D.	WORD PARAMETER	39
35	0000H	2	D.	WORD PARAMETER	36
29	0000H	2	D.	WORD PARAMETER	30
32	0000H	2	D.	WORD PARAMETER	33
193	01EDH	100	DATABYTE.	PROCEDURE STACK=001AH	229 231 232 233
2			DCL.	LITERALLY	
38	0000H		DECOU.	PROCEDURE EXTERNAL(11) STACK=0000H	
81	0000H	1	DEFAULTDRIVE.	BYTE EXTERNAL(86)	
81	0001H	1	DESCR.	BYTE MEMBER(TOKEN)	
76	0000H	1	DEV.	BYTE PARAMETER	77
14	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H	
4			DRCODEDATA.	LITERALLY	84
4			DRCODESEGM.	LITERALLY	84
4			DRDATADATA.	LITERALLY	84
4			DRDATASEGM.	LITERALLY	84
4			DREXTRADATA.	LITERALLY	84
4			DREXTRASEGM.	LITERALLY	84
4			DRSTACKDATA.	LITERALLY	84
4			DRSTACKSEGM.	LITERALLY	84
81	0000H	2	DSCIP.	WORD EXTERNAL(38)	
4			DSDATA.	LITERALLY	
81	0000H	1	DSPEC.	BYTE EXTERNAL(44)	
4			DSVALUE.	LITERALLY	
2			ELSEDD.	LITERALLY	
219	0268H	70	EMITCODEBYTE.	PROCEDURE PUBLIC STACK=0022H	
212	0251H	15	EMITINIT.	PROCEDURE PUBLIC STACK=0002H	
216	0260H	8	EMITTERMINATE.	PROCEDURE PUBLIC STACK=0014H	
84			EMPTY.	LITERALLY	141 153 170 178 189 198 214
2			ENDCASE.	LITERALLY	
2			ENDDD.	LITERALLY	
2			ENDFORLEVER.	LITERALLY	
2			ENDIF.	LITERALLY	
2			ENDMODULE.	LITERALLY	
2			ENDOFFILE.	LITERALLY	
2			ENDPROC.	LITERALLY	
2			ENDWHILE.	LITERALLY	
137	0112H	12	ENTERINPUT.	PROCEDURE STACK=0016H	184 187 205
126	00E5H	45	ENTERNEWBYT.	PROCEDURE STACK=0012H	138 160 161 164
					165 167 168
140	011EH	38	EOFRECORD.	PROCEDURE STACK=0010H	217
81	0000H	1	EOFSET.	BYTE EXTERNAL(70)	
4			EOFTYPE.	LITERALLY	84 143
29	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(8) STACK=0000H	
81	0000H	1	ERRORPRINTED.	BYTE EXTERNAL(73)	
81	0000H	2	ERRORS.	WORD EXTERNAL(63)	
81	0000H	2	ESCIPI.	WORD EXTERNAL(35)	
4			ESDATA.	LITERALLY	
81	0000H	1	ESPEC.	BYTE EXTERNAL(46)	
4			ESVALUE.	LITERALLY	
2			FALSE.	LITERALLY	213

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

8	0000H	2	FCBADR	WORD PARAMETER	9			
8	0000H		FILEADDR	PROCEDURE EXTERNAL(1) STACK=0000H				
3			FILESTRUCTURE . . .	LITERALLY				
3			FILESTRUCTURE . . .	LITERALLY				
79	0000H		FILESFUP	PROCEDURE BYTE EXTERNAL(29) STACK=0000H				
11	0000H		FILL	PROCEDURE EXTERNAL(?) STACK=0000H				
76	0000H	2	FILNAM	WORD PARAMETER	77			
76	0000H	2	FILTYP	WORD PARAMETER	77			
81	0000H	2	FIRSTMACROPTR . . .	WORD EXTERNAL(74)				
2			FOREVER	LITERALLY				
2			FORMFEED	LITERALLY				
81	0000H	1	FULLSYMBTAB	BYTE EXTERNAL(76)				
84	0004H	1	G8YT	BYTE	138	206	221	
82	0000H		GLOBALINIT	PROCEDURE EXTERNAL(89) STACK=0000H				
84	0003H	1	GTP	BYTE	176	180	181	196 200 201 222
81	0000H	5	HELP	BYTE ARRAY(5) EXTERNAL(84)		119	120	121
32	0000H		HEX1OUT	PROCEDURE EXTERNAL(9) STACK=0000H			119	206
35	0000H		HEX2OUT	PROCEDURE EXTERNAL(10) STACK=0000H				
1	0000H		HEXOUT	PROCEDURE STACK=0000H				
81	0000H	1	I	BYTE EXTERNAL(85)		112	113	114 118 119
76	0000H		IFILESETUP	PROCEDURE EXTERNAL(28) STACK=0000H				
81	0000H	1	IFLEVEL	BYTE EXTERNAL(78)				
81	0000H	1	IFLIST	BYTE EXTERNAL(49)				
81	0000H	1	INCLUDEDEFAULT . . .	BYTE EXTERNAL(87)				
81	0000H	1	INCLUDEON	BYTE EXTERNAL(77)				
3			INDEXREGBIT	LITERALLY				
3			INDEXREGCOUNT . . .	LITERALLY				
52	0000H		ININCLUDEBYTE	PROCEDURE BYTE EXTERNAL(16) STACK=0000H				
2			INIT	LITERALLY				
50	0000H		INSOURCEBYTE	PROCEDURE BYTE EXTERNAL(15) STACK=0000H				
4			INTELDATA	LITERALLY	34			
81	0000H	1	INTELHEXON	BYTE EXTERNAL(71)		87		
4			INTELSEGMENT	LITERALLY	84			
3			IREGBIT	LITERALLY				
3			IREGCOUNT	LITERALLY				
81	0000H	2	LENGTH	WORD MEMBER(OPERANDS)				
81	0000H	2	LENGTH	WORD MEMBER(CURRENTSYMBOL)				
17	0000H		LETTER	PROCEDURE BYTE EXTERNAL(4) STACK=0000H				
2			LF	LITERALLY	124			
2			LIT	LITERALLY	2	3	4	84
84	0000H	2	LOCCIP	WORD	196	210		
81	0000H	2	MACROPTR	WORD EXTERNAL(75)				
81	0000H	1	MAXCOL	BYTE EXTERNAL(50)				
11	0000H	1	N	BYTE PARAMETER	12			
38	0000H	2	N	WORD PARAMETER	39			
35	0000H	2	N	WORD PARAMETER	36			
32	0000H	1	N	BYTE PARAMETER	33			
29	0000H	1	N	BYTE PARAMETER	30			
85	0004H	1	N	BYTE PARAMETER AUTOMATIC		86	89	92
81	0000H	1	NEXTCH	BYTE EXTERNAL(65)				
81	0000H	1	NOERROR	BYTE EXTERNAL(72)				
81	0000H	1	NOOPER	BYTE EXTERNAL(61)				
3			NOOVERCOUNT	LITERALLY				
3			NOOVERKIDEBIT	LITERALLY				
81	0006H	2	OFFSET	WORD MEMBER(OPERANDS)				
81	0006H	2	OFFSET	WORD MEMBER(CURRENTSYMBOL)				
84	0008H	2	OFFSETADDR	WORD AT	109	131	144	156 182 202

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

149	0001H	1	OFFSHIGH	BYTE EXTERNAL(31) AT	167				
149	0000H	1	OFFSLOW	BYTE EXTERNAL(31) AT	168				
58	0000H		OPENHEX	PROCEDURE EXTERNAL(19) STACK=0000H					
56	0000H		OPENINCLUDE	PROCEDURE EXTERNAL(18) STACK=0000H					
60	0000H		OPENPRINT	PROCEDURE EXTERNAL(20) STACK=0000H					
54	0000H		OPENSOURCE	PROCEDURE EXTERNAL(17) STACK=0000H					
62	0000H		OPENSYMBOL	PROCEDURE EXTERNAL(21) STACK=0000H					
81	0000H	36	OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(82)					
3			OPERANDSTRUC	LITERALLY	81				
41	0000H		OUTHEXBYTE	PROCEDURE EXTERNAL(12) STACK=0000H		106			
44	0000H		OUTPRINTBYTE	PROCEDURE EXTERNAL(13) STACK=0000H					
47	0000H		OUTSYMBOLBYTE	PROCEDURE EXTERNAL(14) STACK=0000H					
5	0000H		OUTTEXT	PROCEDURE EXTERNAL(0) STACK=0000H					
95	0004H	2	P	WORD PARAMETER AUTOMATIC	96	97	98	99	
				100 101 102					
81	0000H	1	PAGESIZE	BYTE EXTERNAL(56)					
81	0000H	1	PASS	BYTE EXTERNAL(30)	150	174	194		
81	0000H	240	PREFIX	BYTE ARRAY(240) EXTERNAL(60)			206		
81	0000H	1	PREFIXPTR	BYTE EXTERNAL(61)	206	207			
81	0000H	1	PRINTDEVICE	BYTE EXTERNAL(53)					
81	0000H	1	PRINTON	BYTE EXTERNAL(47)					
81	0000H	1	PRINTSWITCHOFF	BYTE EXTERNAL(49)					
2			PROC	LITERALLY	20	23	32	35	38 41 44
				47 50 52 54 56 58 60 62 64 66					
				68 70 72 74 76 79 104 103 126 137					
				140 148 173 193 212 216 219					
11	0000H	2	PT	WORD PARAMETER	12				
84	0007H	1	RECORDLG	BYTE AT	110				
84			RECORDLIMIT	LITERALLY	128				
84	000AH	1	RECORDTYPE	BYTE AT	143	157	181	201	
84	000CH	12	RECTYPOTAB	BYTE ARRAY(12) DATA	92				
84	0000H	12	RECTYPI TAB	BYTE ARRAY(12) DATA	89				
85	0000H	34	RECTYPTAB	PROCEDURE BYTE STACK=0004H		143	157	181	
				201					
2			REENT	LITERALLY					
66	0000H		REWINDSOURCE	PROCEDURE EXTERNAL(23) STACK=0000H					
29	0000H	2	S	WORD PARAMETER	30				
96	002BH	1	S1	BYTE	97	100			
96	002CH	1	S2	BYTE	99	102			
81	0000H	12	SAVESOURCE	BYTE ARRAY(12) EXTERNAL(52)					
149	0001H	1	SEGHIGH	BYTE EXTERNAL(33) AT	160				
149	0000H	1	SEGLOW	BYTE EXTERNAL(33) AT	161				
3			SEGMBIT	LITERALLY					
173	01A9H	68	SEGMBYTE	PROCEDURE STACK=001AH		234	235	236	237
3			SEGMCOUNT	LITERALLY					
81	0004H	2	SEGMENT	WORD MEMBER(OPERANDS)					
81	0004H	2	SEGMENT	WORD MEMBER(CURRENTSYMBOL)					
3			SEGTYPEBIT	LITERALLY					
3			SEGTYPECOUNT	LITERALLY					
81	0003H	1	SFLAG	BYTE MEMBER(OPERANDS)					
81	0003H	1	SFLAG	BYTE MEMBER(CURRENTSYMBOL)					
81	0000H	1	SINFORM	BYTE EXTERNAL(57)					
81	0000H	80	SOURCEBUF	BYTE ARRAY(80) EXTERNAL(58)					
81	0000H	12	SOURCENAME	BYTE ARRAY(12) EXTERNAL(51)					
81	0000H	1	SOURCEPTR	BYTE EXTERNAL(59)					
2			SPACE	LITERALLY					
81	0000H	2	SSCTP	WORD EXTERNAL(37)					

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

4			SSDATA	LITERALLY	
81	0000H	1	SSPEC.	BYTE EXTERNAL(45)	
4			SSVALUE.	LITERALLY	
84	0002H	1	STARTFOUND.	BYTE	152 213 225
148	0144H	101	STARTRECORD.	PROCEDURE STACK=0016H	227
4			STARTYPE.	LITERALLY	84 157
2			STRUC.	LITERALLY	31
81	0002H	1	STYPE.	BYTE MEMBER(OPERANDS)	
81	0002H	1	STYPE.	BYTE MEMBER(CURRENTSYMBOL)	
84	0006H	1	SUM.	BYTE	111 114 115
95	0022H	42	SWITCHHIGHLOW.	PROCEDURE STACK=0004H	109
81	0000H	1	SYMBOLDEVICE	BYTE EXTERNAL(54)	
3			SYMBOLHEAD	LITERALLY	
3			SYMBOLSTRUC.	LITERALLY	81
81	0000H	2	SYMBTADDR	WORD EXTERNAL(80)	
5	0000H	2	T.	WORD PARAMETER	6
2			TAB.	LITERALLY	
8	0000H	2	TEXTADR.	WORD PARAMETER	9
2			THENDO	LITERALLY	37 128 150 158 174 176 194
					196 225
81	0000H	30	TITLE.	BYTE ARRAY(30) EXTERNAL(55)	
81	0000H	4	TOKEN.	STRUCTURE EXTERNAL(64)	
2			TRUE	LITERALLY	152
219	0004H	1	TYP.	BYTE PARAMETER AUTOMATIC	220 222 223
81	0000H	1	TYPE	BYTE MEMBER(TOKEN)	
3			TYPEBIT.	LITERALLY	
3			TYPECOUNT.	LITERALLY	
26	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(7) STACK=0000H	
81	0002H	2	VALUE.	WORD MEMBER(TOKEN)	
104	004CH	13	WRITERYT	PROCEDURE STACK=0008H	117 120 121 123
					124
108	0059H	140	WRITERECORD.	PROCEDURE STACK=000CH	130 142 146 154
					169 179 188 199

MODULE INFORMATION:

CODE AREA SIZE = 02AEH 686D
 CONSTANT AREA SIZE = 0018H 24D
 VARIABLE AREA SIZE = 002DH 45D
 MAXIMUM STACK SIZE = 0022H 34D
 611 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE EXPRES

OBJECT MODULE PLACED IN EXPR.OBJ

COMPILER INVOKED BY: :F0: EXPR.PLH DEBUG OPTIMIZE(2) DATE(10/27/81) PAGEWIDTH:(100) XREF

1 \$title ("EXPRESSION MODULE")

expres:

do:

/*

modified 4/8/81 R. Silberstein

modified 4/24/81 R. Silberstein

modified 8/19/81 R. Silberstein

*/

/*

This is the module to evaluate expressions and
instruction operands. The entry subroutines are:

EXPRESSION (resultfield) byte

OPERAND byte

The expression subroutine evaluates a numeric or
memory expression. The "operand" routine evaluates
a single instruction operand. Both routines return
FALSE if an error is found, otherwise true.

*/

\$include (:f1:macro.lit)

= \$nolist

\$include (:f1:expr.x86)

= \$nolist

\$include (:f1:ermod.ext)

= \$nolist

\$include (:f1:exglob.ext)

= \$nolist

\$INCLUDE (:F1:SUBR2.EXT)

= \$nolist

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$eject
/***** global variables: *****/
```

69 1

```
dcl
maxlev        lit '5', /* max no of nested parenthesis */
parlevel      byte,    /* current no of parenthesis level */
stck(600)     byte,    /* local stack within module */
savestack     addr,    /* save of initial entry stack */
expresserr    byte,    /* error flag */
noforward     byte,    /* true if undefined symbols to be neglected */
bracketlegal  byte,    /* true if bracket expression is legal */
udefflag      byte;    /* true if an undefined element found */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$object
$include (:f1:bnf.tex)

=
= /***** "BNF"-expression syntax *****/
= /*
= E::= E xor A || E or A || A
= A::= A and N || N
= N::= not N || R
= R::= P eq P || P lt P || P le P || P gt P || P ge P || P ne P || P
= P::= P + T || P - T || T
= T::= T * M || T / M || T mod M || T shl M || T shr M || R
= M::= - M || + M || S
= S::= <over>: F || F
= F::= F ptr B || seg B || offset B || type B ||
=        length B || last B || B
= B::= ( E ) || [ bracket-expression ] || I
= I::= variable || . number || number || label || string
= <over>::= segment register
= (stringlength < 3)
=
= */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$eject
/***** miscellaneous subroutines: *****/
```

```
70 1  expexit: proc (dummy);
71 2      dcl dummy byte;
72 2      stackptr=savestack;
73 2  end expexit;
```

```
74 1  errorexit: proc;      /* return if wrong syntax */
75 2      dcl dummy byte at (.udefflag);
76 2      expresserr=false;
77 2      call expexit(dummy);
78 2  end errorexit;
```

```
79 1  clearoperand: proc(p);
80 2      dcl p address,oper based p operandstruc;
81 2      CALL FILL (0, .OPER.BASEINDEX - .OPER + 1, P);
82 2      OPER.BASEINDEX = NOOVERRIDEBIT;
83 2  end clearoperand;
```

```
/* routine to test if current token is member of a given
set of special characters.
```

```
Entry parameters: base = exitvalue if token is 1. member of set
                  numbel = no of elements in set
                  pt = pointer to list of elements
Exit value:
routine= 0ffh if token not member of list
         base+i if token is element i,
         token is skipped */
```

```
84 1  specmember: proc (base,numbel,pt) byte;
85 2      dcl (base,numbel,i) byte,pt address,list based pt (1) byte;
86 2      i=0ffh;
87 2      do while (i:=i+1) < numbel;
88 3          if specialtoken(list(i)) then$do call scan; return base+i; end$if;
          end$while;
94 2      return 0ffh;
95 2  end specmember;
```

```
/* Routine to test if current token is member of a given set of
operators.
```

```
Entry/exit : see "specmember" header */
```

```
96 1  opmember: proc(base,numbel,pt) byte;
97 2      dcl (base,numbel,i,byteval) byte,pt address,list based pt (1) byte;
98 2      if token.type = operator then$do
100 3          i=0ffh;
101 3          do while (i:=i+1) < numbel;
102 4              byteval=token.value;
103 4              if byteval=list(i) then$do call scan; return base+i; end$if;
              end$while;
109 3          end$if;
110 2          return 0ffh;
111 2  end opmember;
```

```
/* test if both operands are numbers, if not, error */
```

```
112 1  numbtest: proc (pt1,ptr);
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

113 2      dcl (ptl,ptr) address,(left based ptl,righ based ptr) operandstruc;
114 2      if (left.stype <> number) or (righ.stype <> number) then
115 2          call errexit;
116 2      end numtest;

/* find resulting symbol type as result of an addition or a
   subtraction, test if illegal types */
117 1      typefind: proc (ptl,ptr);
118 2          dcl (ptl,ptr) address,type byte,
              (left based ptl,righ based ptr) operandstruc;
119 2          dcl err lit '07fh',
              crosstab(9) byte data(number,variable,lab,variable,err,err,
                                      lab,err,err);

120 2          typeno: proc(typ) byte;
121 3              dcl typ byte;
122 3              if typ=number then return 0;
124 3              if typ=variable then return 1;
126 3              if typ=lab then return 2;
128 3              call errexit; /* illegal member of expression */
129 3          end typeno;

130 2          stype=crosstab(typeno(left.stype)*3+typeno(righ.stype));
131 2          if stype=err then call errexit;
133 2          left.length=left.length+righ.length;
134 2          left.stype=stype;
135 2      end typefind;

/* take care of segment specification in front of variables
   syntax: <over>: variable, <over>=ES/SS/DS/CS */
136 1      segover: proc(pt) byte;
137 2          dcl pt address,segreg based pt byte;
138 2          if (token.type=reg) and (token.descr=dword) then$do
140 3              if nextch=':' then$do
142 4                  segreg=token.value;
143 4                  segreg=(shl(segreg,segtypecount) and segtypebit) or segmbit;
144 4                  call scan; /* skip segment register */
145 4                  call scan; /* skip : */
146 4                  return 0;
147 4              end$if;
            end$if;
149 2          return 0ffh;
150 2      end segover;

/* create a number operator */
151 1      createnumber: proc(p,n);
152 2          dcl p address,n addr,oper based p operandstruc;
153 2          call clearoperand(.oper);
154 2          oper.stype=number;
155 2          oper.offset=n;
156 2      end createnumber;

/* get current identifier, perform symboltable lookup
   set undefined-symbol-flag if symbol not defined,
   treat undefined symbols as numbers */

157 1      finditem: proc (pt);
158 2          dcl pt address,left based pt operandstruc,symbptr address,i byte;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

159 2      if token.type <> ident then$do
161 3          call clearoperand(.left);
162 3          left.stype=token.type;
163 3          left.sflag=token.descr;
164 3          left.offset=token.value;
165 3      else$do
167 3          if findsymbol(acclen,.accum(0),.symptr) then$do
169 4              call getattributes(symptr,.left);
170 4              i=left.stype;
171 4              if (i=neglected) or (i=doubleddefined) or (i=udefsymb) then$do
173 5                  udefflag=true;
174 5                  left.stype=number;
175 5                  expresserr=false;
176 5                  call errmsg(udefsymbol);
177 5              end$if;
            else$do

            /* symbol undefined - test if it is to be "neglected" */
180 4          expresserr=false;
181 4          if noforward then$do
183 5              if not newsymbol(acclen,.accum,.symptr) then$do
185 6                  call errorexit;
186 6              end$if;
187 5              left.stype=neglected;
188 5              call enterattributes(symptr,.left);
189 5              end$if;

190 4          call errmsg(udefsymbol);
191 4          udefflag=true;
192 4          end$if;
193 3          end$if;
194 2          call scan;
195 2      end finditem;

/* recognize the different symboltypes for the if (identicator)
   subroutine */
196 1      symtyp: proc(pt) byte;
197 2          dcl pt address, left based pt operandstruc,i byte;
198 2          if specialtoken('$') then return 0;
200 2          if specialtoken('.') then return 1;
202 2          if token.type=string then$do
204 3              if (acclen > 0) and (acclen < 3 ) then return 2;
206 3              return 4;          /* error */
207 3          end$if;
208 2          call finditem(.left);
209 2          i=left.stype;
210 2          if (i=pseudo) or (i=operator) or (i=spec) then return 4; /* error */
212 2          return 3;
213 2      end symtyp;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
select
```

```
/****** subroutines for each "NON-TERMINAL" *****/
/****** in "BNF" syntax *****/
```

```
214 1  II: proc (pt) reentrant;
215 2    dcl pt address, left based pt operandstruc,
216 2    doublebyt addr at (.accum(0)), saved byte;
217 2    do case syntyp(.left);
218 3      do; /* $ */
219 4      left.stype=lab;
220 4      left.sflag=wrđ;
221 4      left.offset=cip;
222 4      if csegspec then$do /* pick up current segment specification */
223 5      left.sflag=shl(csegtyp, segtypecount) or segmbit or wrđ;
224 5      left.segment=csegvalue;
225 5      end$if;
226 4      call scan; /* skip $ */
227 4      end;
228 3      do; /* . number */
229 4      call scan; /* skip . */
230 4      call finditem(.left);
231 4      if left.stype <> number then call errorexit;
232 4      left.stype=variable;
233 4      left.segment=curdseg;
234 4      left.sflag=shl(rds, segtypecount) and segtypebit;
235 4      if dspec then left.sflag=left.sflag or segmbit;
236 4      end;
237 3      do; /* string */
238 4      if acclen=1 then$do
239 5      call createnumber(.left, accum(0));
240 5      else$do
241 5      saveb=accum(0);
242 5      accum(0)=accum(1);
243 5      accum(1)=saveb;
244 5      call createnumber(.left, doublebyt);
245 5      end$if;
246 4      call scan; /* skip string */
247 4      end;
248 3      do; end; /* number, label, variable, register */
249 3      call errorexit;
250 3      end$case;
251 2      end II;
252 1  BB: proc (pt) reentrant;
253 2    dcl pt address, left based pt operandstruc;
254 2    if specialtoken("(") then$do
255 3    if (parlevel:=parlevel+1) > maxlev-1 then call errorexit;
256 3    call scan;
257 3    call EE(.left);
258 3    if not specialtoken(")") then call errorexit;
259 3    parlevel=parlevel-1;
260 3    call scan;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

269 3      return;
270 3      end$if;
271 2      if specialtoken(leftbracket) then$do
273 3          if not bracketlegal then call errorexit;
275 3          bracketlegal=false;
276 3          call scan; /* skip leftbracket */
277 3          call clearoperand(.left);
278 3          left.stype=number;
279 3          if not bracketexpr(.left) then call errorexit;
281 3          return;
282 3      end$if;
283 2      call li(.left);
284 2      end BB;

285 1      FF: proc (pt) reentrant;
286 2          decl pt address, left based pt operandstruc, right operandstruc,
                opertyp byte, val addr;
287 2          if (opertyp:=opmember(0,5,.(oseg,offset,otype,olength,olast)))
                <> 0ffn then$do
289 3              call BB(.left);
290 3              do case opertyp;

291 4                  do; /* SEG */
292 5                      if (left.sflag and segmbit) = 0 then call errorexit;
294 5                      call createnumber(.left, left.segment);
295 5                      end;

296 4                  do; /* OFFSET */
297 5                      call createnumber(.left, left.offset);
298 5                      end;

299 4                  do; /* TYPE */
300 5                      call createnumber(.left, left.sflag and typebit);
301 5                      end;

302 4                  do; /* LENGTH */
303 5                      call createnumber(.left, left.length);
304 5                      end;

305 4                  do; /* LAST */
306 5                      if (val:=left.length) = 0 then val=1;
308 5                      call createnumber(.left, val-1);
309 5                      end;

310 4              end$case;
311 3          else$do
313 3              call BB(.left);
314 3              do while opmember(0,1,.(optr)) <> 0ffh;
315 4                  call Bb(.right);
316 4                  left.stype=right.stype;
317 4                  left.segment=right.segment;
318 4                  left.offset=right.offset;
319 4                  left.baseindex=right.baseindex;
320 4                  left.sflag=(left.sflag and typebit) or (right.sflag and
                (not typebit));

321 4              end$while;
322 3          end$if;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
323  2      end FF;

324  1      SS: proc (pt) reentrant;
325  2          dcl pt address, left based pt operandstruc, segreg byte;
326  2          if segover(segreg) <> 0ffh then do
327  3              call FF(.left);
328  3              left.sflag=(left.sflag and (not segtypebit)) or segreg;
329  3              left.baseindex=left.baseindex and (not nooverridebit);
330  3          elseif do
331  3              call FF(.left);
332  3          endif;
333  2      end SS;

336  1      MM: proc (pt) reentrant;
337  2          dcl pt address, left based pt operandstruc, opertyp byte;
338  2          if (opertyp:=specmember(0,2,.( "+-")) <> 0ffh then do
339  3              call MM(.left);
340  3              call numtest(.left,.left);
341  3              if opertyp=1 then do
342  4                  left.offset=-left.offset;
343  4              endif;
344  3              elseif do
345  3                  call SS(.left);
346  3              endif;
347  2      end MM;

351  1      IT: proc (pt) reentrant;
352  2          dcl pt address, left based pt operandstruc, righth operandstruc,
353  2              opertyp byte, (leftval,righthval) addr;
354  2          call MM(.left);
355  2          do while (opertyp:=specmember(0,2,.( "*/")) and
356  2              opmember(2,3,.( omod,oshl,oshr))) <> 0ffh;
357  3              call MM(.righth);
358  3              call numtest(.left,.righth);
359  3              leftval=left.offset;
360  3              righthval=righth.offset;
361  3              do case opertyp;
362  4                  leftval=leftval*righthval;
363  4                  leftval=leftval/righthval;
364  4                  leftval=leftval mod righthval;
365  4                  if righthval>0 and righthval<16 then leftval=shl(leftval,righthval);
366  4                  if righthval>0 and righthval<16 then leftval=shr(leftval,righthval);
367  4              endcase;
368  3              left.offset=leftval;
369  3          endwhile;
370  2      end IT;

371  1      PP: proc (pt) reentrant;
372  2          dcl pt address, left based pt operandstruc, righth operandstruc,
373  2              opertyp byte;
374  2          call IT(.left);
375  2          do while (opertyp:=specmember(0,2,.( "+-")) <> 0ffh;
376  3              call IT(.righth);
377  3              call typefind(.left,.righth);
378  3              if opertyp=0 then do
379  4                  left.offset=left.offset+righth.offset;
380  4              elseif do
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

382 4      left.offset=left.offset-rigth.offset;
383 4      end$if;
384 3      end$while;
385 2      end PP;

386 1      RR: proc (pt) reentrant;
387 2      dcl pt address, left based pt operandstruc, rigth operandstruc,
          opertyp byte, (leftval, rigthval) addr;
388 2      call PP(.left);
389 2      if (opertyp:=opmember(0,6,.(oeq,olt,ole,ogt,oge,one))) <> 0ffh
          then$do

391 3          call PP(.rigth);
392 3          call numtest(.left,.rigth);
393 3          leftval=left.offset;
394 3          rigthval=rigth.offset;
395 3          do case opertyp;
396 4              leftval = (leftval = rigthval);
397 4              leftval = (leftval < rigthval);
398 4              leftval = (leftval <= rigthval);
399 4              leftval = (leftval > rigthval);
400 4              leftval = (leftval >= rigthval);
401 4              leftval = (leftval <> rigthval);
402 4          end$case;
403 3          IF LEFTVAL = OFFH THEN LEFTVAL = OFFFH;
405 3          left.offset=leftval;
406 3          end$if;
407 2      end RR;

408 1      NN: proc (pt) reentrant;
409 2      dcl pt address, left based pt operandstruc;
410 2      if opmember(0,1,.(onot)) <> 0ffh then$do
412 3          call NN(.left);
413 3          call numtest(.left,.left);
414 3          left.offset=not left.offset;
415 3      else$do
417 3          call RR(.left);
418 3      end$if;
419 2      end NN;

420 1      AA: proc (pt) reentrant;
421 2      dcl pt address, left based pt operandstruc, rigth operandstruc;
422 2      call NN(.left);
423 2      do while opmember(0,1,.(oand)) <> 0ffh;
424 3          call NN(.rigth);
425 3          call numtest(.left,.rigth);
426 3          left.offset=left.offset and rigth.offset;
427 3      end$while;
428 2      end AA;

429 1      EE: proc (pt) reentrant;
430 2      dcl pt address, left based pt operandstruc, right operandstruc,
          opertype byte;

431 2      call AA(.left);
432 2      do while (opertype:=opmember(0,2,.(oor,oxor))) <> 0ffh;
433 3          call AA(.right);
434 3          call numtest(.left,.right);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
435 3      if opertype=0 then$do
437 4          left.offset=left.offset or right.offset;
438 4      else$do
440 4          left.offset=left.offset xor right.offset;
441 4      end$if;
442 3      end$while;
443 2  end EE;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$object
/***** MATH SUBROUTINES *****/
```

```
444 1  realexpress: proc(pt);
445 2      dcl pt address,oper based pt operandstruc,
          dummy byte at(.udefflag);
446 2      savestack=stackptr; /* use local stack for reentrant routines */
447 2      stackptr=.stck(length(stck));
448 2      call EEC(oper);
449 2      call expexit(dummy);
450 2  end realexpress;
```

```
451 1  express: proc(pt) byte;
452 2      dcl pt address,oper based pt operandstruc;
453 2      expresserr=true;
454 2      udefflag=false;
455 2      parlevel=0;
456 2      call realexpress(.oper);
457 2      if udefflag then$do
459 3          oper.stype=number;
460 3          oper.sflag=byt;
461 3          oper.offset=0;
462 3      end$if;
463 2      return expresserr;
464 2  end express;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

```
/* normal expression */
```

```
465 1  expression: proc (pt) byte public;
466 2      dcl pt address;
467 2      noforward=false;
468 2      bracketlegal=false;
469 2      return express(pt);
470 2  end expression;
```

```
/* special expression - mark all undefined symbols as "neglected" */
```

```
471 1  noforwardexpr: proc(pt) byte public;
472 2      dcl pt address;
473 2      noforward=true;
474 2      bracketlegal=false;
475 2      return express(pt);
476 2  end noforwardexpr;
```

```
477 1  OPERND; PROC BYTE;
478 2      dcl exitvalue byte,pt address,oper based pt operandstruc;

479 2      pt=.operands(nooper);
480 2      exitvalue=true;
481 2      bracketlegal=true;
482 2      exitvalue=express(pt);
483 2      if specialtoken(leftbracket) then$do
485 3          if bracketlegal then$do
487 4              call scan;
488 4              exitvalue=exitvalue and bracketexpr(pt);
489 4          else$do
491 4              exitvalue=false;
```



```
492 4      endif;
493 3      endif;
494 2      return exitvalue;
495 2      END OPERND;

496 1      OPERAND: PROC BYTE PUBLIC;
497 2          NOFORWARD = FALSE;
498 2          RETURN OPERND;
499 2      END OPERAND;

500 1      NOFORWARDOPER: PROC BYTE PUBLIC;
501 2          NOFORWARD = TRUE;
502 2          RETURN OPERND;
503 2      END NOFORWARDOPER;

504 1      end$module expres;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
420	07EBH	70	AA PROCEDURE REENTRANT STACK=0036H 431 433
32	0000H	1	ACCLEN BYTE EXTERNAL(16) 167 183 204 240
32	0000H	80	ACCUM. BYTE ARRAY(80) EXTERNAL(17) 167 183 215
			242 245 246 247
2			ADDR LITERALLY 10 13 16 19 32 34 40
			58 61 64 67 69 80 113 118 152 158
			197 215 253 236 325 337 352 372 387 409
			421 430 445 452 478
48	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(25) STACK=0000H
51	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(26) STACK=0000H
96	0008H	1	BASE BYTE PARAMETER AUTOMATIC 97 106
84	0008H	1	BASE BYTE PARAMETER AUTOMATIC 85 91
32	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
452	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
158	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
352	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
113	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
113	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
152	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
215	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
352	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
478	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
430	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
372	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
421	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
387	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
421	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
430	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
409	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
286	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT) 319
372	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
258	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
337	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
118	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
286	0008H	1	BASEINDEX. BYTE MEMBER(LEFT) 319
445	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
325	0008H	1	BASEINDEX. BYTE MEMBER(LEFT) 330
387	0008H	1	BASEINDEX. BYTE MEMBER(RIGHT)
118	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
80	0008H	1	BASEINDEX. BYTE MEMBER(OPER) 81 82
197	0008H	1	BASEINDEX. BYTE MEMBER(LEFT)
7			BASEREGIT LITERALLY
7			BASEREGCOUNT LITERALLY
257	0380H	143	BB PROCEDURE REENTRANT STACK=0026H 289 313
			315
26	0000H		BRACKETEXPR. PROCEDURE BYTE EXTERNAL(6) STACK=0000H 279
			488
69	0261H	1	BRACKETLEGAL BYTE 273 275 468 474 481 485
7			BREGBIT. LITERALLY

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

7		BRECCOUNT.	LITERALLY				
4		BYT.	LITERALLY	450			
97	0265H	1 BYTEVAL.	BYTE	192	103		
45	0000H	1 CH.	BYTE PARAMETER		46		
48	0000H	1 CH.	BYTE PARAMETER		49		
51	0000H	1 CH.	BYTE PARAMETER		52		
54	0000H	1 CH.	BYTE PARAMETER		55		
39	0000H	1 CH.	BYTE PARAMETER		47		
42	0000H	1 CH.	BYTE PARAMETER		43		
32	0000H	2 CIP.	WORD EXTERNAL(8)		220		
79	001DH	33 CLEAROPERAND.	PROCEDURE STACK=000CH		153	151	277
3		CODE.	LITERALLY				
2		CR.	LITERALLY				
151	0184H	32 CREATENUMBER.	PROCEDURE STACK=0014H		242	248	294 297
			300 303 308				
119	0000H	9 CROSSTAR.	BYTE ARRAY(9) DATA		130		
32	0000H	1 CSEGSPEC.	BYTE EXTERNAL(11)		221		
32	0000H	1 CSECTYPE.	BYTE EXTERNAL(9)		223		
32	0000H	2 CSEGVALUE.	WORD EXTERNAL(10)		224		
32	0000H	2 CRODSEG.	WORD EXTERNAL(13)		234		
63	0000H	2 D.	WORD PARAMETER		64		
57	0000H	2 D.	WORD PARAMETER		58		
66	0000H	2 D.	WORD PARAMETER		67		
60	0000H	2 D.	WORD PARAMETER		61		
2		DCL.	LITERALLY				
66	0000H	DECOUT.	PROCEDURE EXTERNAL(31) STACK=0000H				
3		DELETEDSYMB.	LITERALLY				
32	0001H	1 DESCR.	BYTE MEMBER(TOKEN)		138	163	
15	0000H	2 DEST.	WORD PARAMETER		16		
42	0000H	DIGIT.	PROCEDURE BYTE EXTERNAL(23) STACK=0000H				
215	0000H	2 DOUBLEBYT.	WORD EXTERNAL(17) AT		248		
3		DOUBLEDEFINED.	LITERALLY		171		
32	0000H	1 DSPEC.	BYTE EXTERNAL(12)		236		
70	0004H	1 DUMMY.	BYTE PARAMETER AUTOMATIC		71		
75	0262H	1 DUMMY.	BYTE AT		77		
445	0262H	1 DUMMY.	BYTE AT		449		
4		DWRD.	LITERALLY		138		
429	0831H	90 EE.	PROCEDURE RECENTRANT STACK=0046H		264	443	
2		ELSEDO.	LITERALLY		177	345	
2		ENDCASE.	LITERALLY		366		
2		ENDDO.	LITERALLY				
2		ENDFOREVER.	LITERALLY				
2		ENDIF.	LITERALLY		147		
2		ENDMODULE.	LITERALLY				
2		ENDOFFILE.	LITERALLY				
2		ENDPROC.	LITERALLY				
2		ENDWHILE.	LITERALLY		92	107	
18	0000H	ENTERATTRIBUTES.	PROCEDURE EXTERNAL(3) STACK=0000H			188	
57	0000H	EQUAL.	PROCEDURE BYTE EXTERNAL(28) STACK=0000H				
119		ERR.	LITERALLY		119	131	
29	0000H	ERRMSG.	PROCEDURE EXTERNAL(7) STACK=0000H			176	190
29	0000H	1 ERRNO.	BYTE PARAMETER		30		
74	000CH	17 ERROREXIT.	PROCEDURE STACK=0008H		115	128	132 185
			232 254 262 266 274 280 293				
478	0269H	1 EXITVALUE.	BYTE		480	482	488 491 494
1	0000H	EXPRES.	PROCEDURE STACK=0000H				
451	0880H	63 EXPRESS.	PROCEDURE BYTE STACK=0032H		469	475	482

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

69	025FH	1	EXPRESSERR	BYTE	76	175	180	453	463						
465	03EFH	23	EXPRESSION	PROCEDURE	BYTE PUBLIC STACK=0028H										
70	0000H	12	EXPEXIT	PROCEDURE	STACK=0004H			77	449						
2			FALSE	LITERALLY		76	175	180	275	454	467	463			
					474	491	497								
36	0000H	2	FCBADR	WORD	PARAMETER			37							
285	043FH	247	FF	PROCEDURE	REENTRANT STACK=0038H					328	333				
36	0000H		FILEABORT	PROCEDURE	EXTERNAL(21) STACK=0000H										
39	0000H		FILL	PROCEDURE	EXTERNAL(22) STACK=0000H					81					
157	01A4H	200	FINDITEM	PROCEDURE	STACK=0012H			208	230						
12	0000H		FINDSYMBOL	PROCEDURE	BYTE EXTERNAL(1) STACK=0000H					167					
2			FOREVER	LITERALLY											
2			FORMFIELD	LITERALLY											
15	0000H		GETATTRIBUTES	PROCEDURE	EXTERNAL(2) STACK=0000H					169					
60	0000H		HEX1OUT	PROCEDURE	EXTERNAL(29) STACK=0000H										
63	0000H		HEX2OUT	PROCEDURE	EXTERNAL(30) STACK=0000H										
97	0264H	1	I	BYTE	100	101	103	106							
158	0267H	1	I	BYTE	170	171									
85	0263H	1	I	BYTE	86	87	88	91							
197	0268H	1	I	BYTE	209	210									
3			IDENT	LITERALLY		159									
214	02CCH	228	II	PROCEDURE	REENTRANT STACK=0020H					283					
7			INDEXREGBIT	LITERALLY											
7			INDEXREGCOUNT	LITERALLY											
2			INIT	LITERALLY											
7			IREGBIT	LITERALLY											
7			IREGCOUNT	LITERALLY											
3			LAB	LITERALLY		119	126	218							
158	0000H	9	LEFT	STRUCTURE	BASED(PT)		161	162	163	164	169				
					170	174	187	188							
409	0000H	9	LEFT	STRUCTURE	BASED(PT)		412	413	414	417					
421	0000H	9	LEFT	STRUCTURE	BASED(PT)		422	425	426						
215	0000H	9	LEFT	STRUCTURE	BASED(PT)		216	218	219	220	223				
					224	230	231	233	234	235	237	242	248		
352	0000H	9	LEFT	STRUCTURE	BASED(PT)		353	356	357	368					
372	0000H	9	LEFT	STRUCTURE	BASED(PT)		373	376	379	382					
337	0000H	9	LEFT	STRUCTURE	BASED(PT)		340	341	344	346					
258	0000H	9	LEFT	STRUCTURE	BASED(PT)		264	277	278	279	283				
430	0000H	9	LEFT	STRUCTURE	BASED(PT)		431	434	437	440					
387	0000H	9	LEFT	STRUCTURE	BASED(PT)		388	392	393	405					
197	0000H	9	LEFT	STRUCTURE	BASED(PT)		208	209							
325	0000H	9	LEFT	STRUCTURE	BASED(PT)		328	329	330	333					
118	0000H	9	LEFT	STRUCTURE	BASED(PTL)		130	133	134						
286	0000H	9	LEFT	STRUCTURE	BASED(PT)		289	292	294	297	300				
					303	306	308	313	316	317	318	319	320		
113	0000H	9	LEFT	STRUCTURE	BASED(PTL)		114								
6			LEFTBRACKET	LITERALLY		271	493								
387	0002H	2	LEFTVAL	WORD	AUTOMATIC		393	396	397	398	399	400			
					401	403	404	405							
352	0002H	2	LEFTVAL	WORD	AUTOMATIC		357	360	361	362	364	366			
					368										
421	0000H	2	LENGTH	WORD	MEMBER(RIGHT)										
325	0000H	2	LENGTH	WORD	MEMBER(LEFT)										
372	0000H	2	LENGTH	WORD	MEMBER(LEFT)										
409	0000H	2	LENGTH	WORD	MEMBER(LEFT)										
372	0000H	2	LENGTH	WORD	MEMBER(RIGHT)										
421	0000H	2	LENGTH	WORD	MEMBER(LEFT)										

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

352	0000H	2	LENGTH	WORD MEMBER(RIGHT)	
337	0000H	2	LENGTH	WORD MEMBER(LEFT)	
352	0000H	2	LENGTH	WORD MEMBER(LEFT)	
430	0000H	2	LENGTH	WORD MEMBER(RIGHT)	
387	0000H	2	LENGTH	WORD MEMBER(LEFT)	
32	0000H	2	LENGTH	WORD MEMBER(COPIES)	
286	0000H	2	LENGTH	WORD MEMBER(RIGHT)	
286	0000H	2	LENGTH	WORD MEMBER(LEFT)	303 306
430	0000H	2	LENGTH	WORD MEMBER(LEFT)	
258	0000H	2	LENGTH	WORD MEMBER(LEFT)	
215	0000H	2	LENGTH	WORD MEMBER(LEFT)	
			LENGTH	BUILTIN	447
80	0000H	2	LENGTH	WORD MEMBER(COPIES)	
197	0000H	2	LENGTH	WORD MEMBER(LEFT)	
452	0000H	2	LENGTH	WORD MEMBER(COPIES)	
158	0000H	2	LENGTH	WORD MEMBER(LEFT)	
152	0000H	2	LENGTH	WORD MEMBER(COPIES)	
387	0000H	2	LENGTH	WORD MEMBER(RIGHT)	
118	0000H	2	LENGTH	WORD MEMBER(RIGHT)	133
118	0000H	2	LENGTH	WORD MEMBER(LEFT)	133
478	0000H	2	LENGTH	WORD MEMBER(COPIES)	
113	0000H	2	LENGTH	WORD MEMBER(RIGHT)	
445	0000H	2	LENGTH	WORD MEMBER(COPIES)	
113	0000H	2	LENGTH	WORD MEMBER(LEFT)	
45	0000H		LETTER	PROCEDURE BYTE EXTERNAL(24) STACK=0000H	
2			LF	LITERALLY	
12	0000H	1	LG	BYTE PARAMETER	13
9	0000H	1	LG	BYTE PARAMETER	10
85	0000H	1	LIST	BYTE BASED(PT) ARRAY(1)	88
97	0000H	1	LIST	BYTE BASED(PT) ARRAY(1)	103
2			LIT	LITERALLY	2 3 4 5 6 7 8
					69 119
69			HAXLEV	LITERALLY	261
336	0573H	71	MM	PROCEDURE REENTRANT STACK=0048H	340 353
					355
66	0000H	2	N.	WORD PARAMETER	67
63	0000H	2	N.	WORD PARAMETER	64
39	0000H	1	N.	BYTE PARAMETER	40
57	0000H	1	N.	BYTE PARAMETER	58
60	0000H	1	N.	BYTE PARAMETER	61
151	0004H	2	N.	WORD PARAMETER AUTOMATIC	152 155
3			NEGLECTED	LITERALLY	171 187
9	0000H		NEWSYMBOL	PROCEDURE BYTE EXTERNAL(0) STACK=0000H	183
32	0000H	1	NEXTCH	BYTE EXTERNAL(15)	140
4			NIL	LITERALLY	
408	07AFH	60	NN	PROCEDURE REENTRANT STACK=0086H	412 422
					424
69	0260H	1	NOFORWARD	BYTE	181 467 473 497 501
471	0906H	23	NOFORWARDEXPR	PROCEDURE BYTE PUBLIC STACK=0000H	
500	0979H	13	NOFORWARDOPER	PROCEDURE BYTE PUBLIC STACK=003AH	
32	0000H	1	NOOPER	BYTE EXTERNAL(18)	479
7			NOOVERRIDE	LITERALLY	
7			NOOVERRIDEBIT	LITERALLY	82 330
96	0006H	1	NUMBER	BYTE PARAMETER AUTOMATIC	97 101
84	0006H	1	NUMBER	BYTE PARAMETER AUTOMATIC	85 37
3			NUMBER	LITERALLY	114 119 122 154 174 231 278
					459

CCP/M-86 2-0 V.3

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104

112	0086H	28	NUMTEST	PROCEDURE STACK=0010H	341	356	392	413
				425 434				
6			UAND	LITERALLY 423				
6			DEQ.	LITERALLY 339				
452	0006H	2	OFFSET	WORD MEMBER(OPER)	461			
445	0006H	2	OFFSET	WORD MEMBER(OPER)				
430	0006H	2	OFFSET	WORD MEMBER(LEFT)	437	440		
421	0006H	2	OFFSET	WORD MEMBER(RIGHT)	425			
421	0006H	2	OFFSET	WORD MEMBER(LEFT)	425			
409	0006H	2	OFFSET	WORD MEMBER(LEFT)	414			
387	0006H	2	OFFSET	WORD MEMBER(RIGHT)	394			
387	0006H	2	OFFSET	WORD MEMBER(LEFT)	393	405		
372	0006H	2	OFFSET	WORD MEMBER(RIGHT)	379	382		
372	0006H	2	OFFSET	WORD MEMBER(LEFT)	379	382		
352	0006H	2	OFFSET	WORD MEMBER(RIGHT)	358			
352	0006H	2	OFFSET	WORD MEMBER(LEFT)	357	368		
337	0006H	2	OFFSET	WORD MEMBER(LEFT)	344			
32	0006H	2	OFFSET	WORD MEMBER(OPERANDS)				
325	0006H	2	OFFSET	WORD MEMBER(LEFT)				
286	0006H	2	OFFSET	WORD MEMBER(RIGHT)	318			
286	0006H	2	OFFSET	WORD MEMBER(LEFT)	297	318		
478	0006H	2	OFFSET	WORD MEMBER(OPER)				
258	0006H	2	OFFSET	WORD MEMBER(LEFT)				
215	0006H	2	OFFSET	WORD MEMBER(LEFT)	220			
430	0006H	2	OFFSET	WORD MEMBER(RIGHT)	437	440		
197	0006H	2	OFFSET	WORD MEMBER(LEFT)				
158	0006H	2	OFFSET	WORD MEMBER(LEFT)	164			
152	0006H	2	OFFSET	WORD MEMBER(OPER)	155			
118	0006H	2	OFFSET	WORD MEMBER(RIGHT)				
118	0006H	2	OFFSET	WORD MEMBER(LEFT)				
80	0006H	2	OFFSET	WORD MEMBER(OPER)				
113	0006H	2	OFFSET	WORD MEMBER(RIGHT)				
113	0006H	2	OFFSET	WORD MEMBER(LEFT)				
6			OGI.	LITERALLY 389				
6			OGT.	LITERALLY 389				
6			OLAST.	LITERALLY 287				
6			OLE.	LITERALLY 389				
6			OLENGTH.	LITERALLY 287				
6			OLT.	LITERALLY 389				
6			OMOD	LITERALLY 354				
6			ONE.	LITERALLY 389				
6			ONOT	LITERALLY 410				
6			OOFFSET.	LITERALLY 287				
6			OOK.	LITERALLY 432				
452	0000H	9	OPER	STRUCTURE BASED(PT)	456	459	460	461
80	0000H	9	OPER	STRUCTURE BASED(P)	81	82		
152	0000H	9	OPER	STRUCTURE BASED(P)	153	154	155	
445	0000H	9	OPER	STRUCTURE BASED(PT)	446			
478	0000H	9	OPER	STRUCTURE BASED(PT)				
496	096CH	13	OPERAND.	PROCEDURE BYTE PUBLIC STACK=00BAH				
32	0000H	36	OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(19)				479
7			OPERANDSTRUC	LITERALLY 32 80 113 118 152 158 197				
				215 258 286 325 337 352 372 387 409 421				
				430 445 452 473				
3			OPERATOR	LITERALLY 98 210				
477	0910H	79	OPERND	PROCEDURE BYTE STACK=0036H	498	502		
387	000FH	1	OPERTYP.	BYTE AUTOMATIC 339 395				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

352	000FH	1	OPERTYP.	BYTE AUTOMATIC	354	359		
337	0002H	1	OPERTYP.	BYTE AUTOMATIC	358	342		
372	000BH	1	OPERTYP.	BYTE AUTOMATIC	374	377		
286	000DH	1	OPERTYP.	BYTE AUTOMATIC	287	299		
430	000BH	1	OPERTYPE	BYTE AUTOMATIC	432	435		
96	0074H	68	OPMEMBER	PROCEDURE BYTE STACK=000AH	287	314	354	
				389 410 423 432				
6			OPTR	LITERALLY	314			
6			OSLG	LITERALLY	287			
6			OSHL	LITERALLY	354			
6			OSHORT	LITERALLY				
6			OSHR	LITERALLY	354			
6			OTYPE	LITERALLY	287			
33	0000H		OUTTEXT.	PROCEDURE EXTERNAL(20) STACK=0000H				
6			OXOR	LITERALLY	432			
79	0004H	2	P.	WORD PARAMETER AUTOMATIC	80	91	82	
151	0006H	2	P.	WORD PARAMETER AUTOMATIC	152	153	154	155
69	0006H	1	PARLEVEL	BYTE	261	267	455	
371	0685H	90	PP	PROCEDURE REENTRANT STACK=000CH			388	391
2			PROC	LITERALLY	9	12	15	18
					21	23	26	
					29	48	51	60
					63	66	70	74
					79	84		
					96	112	117	120
					136	151	157	196
					214	257		
					285	324	336	351
					371	386	408	420
					429	444		
					451	465	471	477
					490	500		
3			PSEUDO	LITERALLY	210			
465	0004H	2	PT	WORD PARAMETER AUTOMATIC	466	469		
84	0004H	2	PT	WORD PARAMETER AUTOMATIC	85	88		
451	0004H	2	PT	WORD PARAMETER AUTOMATIC	452	456	459	460
				461				
429	000EH	2	PT	WORD PARAMETER AUTOMATIC	430	431	434	437
				440				
420	000EH	2	PT	WORD PARAMETER AUTOMATIC	421	422	425	426
408	0004H	2	PT	WORD PARAMETER AUTOMATIC	409	412	413	414
				417				
386	0012H	2	PT	WORD PARAMETER AUTOMATIC	387	388	392	393
				405				
371	000EH	2	PT	WORD PARAMETER AUTOMATIC	372	373	376	379
				382				
39	0000H	2	PT	WORD PARAMETER	40			
351	0012H	2	PT	WORD PARAMETER AUTOMATIC	352	353	356	357
				368				
336	0006H	2	PT	WORD PARAMETER AUTOMATIC	337	340	341	344
				348				
324	0006H	2	PT	WORD PARAMETER AUTOMATIC	325	328	329	330
				333				
285	0010H	2	PT	WORD PARAMETER AUTOMATIC	286	289	292	294
				297	300	303	306	308
				313	316	317	318	319
				320				
257	0004H	2	PT	WORD PARAMETER AUTOMATIC	258	264	277	278
				279	283			
26	0000H	2	PT	WORD PARAMETER	27			
214	0006H	2	PT	WORD PARAMETER AUTOMATIC	215	216	218	219
				220	223	224	230	231
				233	234	235	237	242
				248				
196	0004H	2	PT	WORD PARAMETER AUTOMATIC	197	208	209	
157	0004H	2	PT	WORD PARAMETER AUTOMATIC	158	161	162	163

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

444	0004H	2	PT	164	169	170	174	187	188											
136	0004H	2	PT	WORD	PARAMETER	AUTOMATIC				445	446									
96	0004H	2	PT	WORD	PARAMETER	AUTOMATIC				137	142	143								
478	0004H	2	PT	WORD			479	479	482	433										
471	0004H	2	PT	WORD	PARAMETER	AUTOMATIC				472	475									
117	0006H	2	PTL	WORD	PARAMETER	AUTOMATIC				118	130	133	134							
112	0006H	2	PTL	WORD	PARAMETER	AUTOMATIC				113	114									
117	0004H	2	PTR	WORD	PARAMETER	AUTOMATIC				118	130	133								
112	0004H	2	PTR	WORD	PARAMETER	AUTOMATIC				113	114									
5			RBP	LITERALLY																
5			RBX	LITERALLY																
5			RCS	LITERALLY																
5			RDI	LITERALLY																
5			RDS	LITERALLY																
444	088BH	37	REALXPRESS	PROCEDURE	STACK=00ACH					456										
2			REENT	LITERALLY																
3			REG	LITERALLY																
5			RES	LITERALLY																
12	0000H	2	RESULT	WORD	PARAMETER				13											
9	0000H	2	RESULT	WORD	PARAMETER				10											
430	0002H	9	RIGHT	STRUCTURE	AUTOMATIC					433	434	437	440							
6			RIGHTBRACKET	LITERALLY																
421	0002H	9	RIGH	STRUCTURE	AUTOMATIC					424	425	426								
387	0006H	9	RIGH	STRUCTURE	AUTOMATIC					391	392	394								
372	0002H	9	RIGH	STRUCTURE	AUTOMATIC					375	376	379	382							
352	0006H	9	RIGH	STRUCTURE	AUTOMATIC					355	356	358								
286	0004H	9	RIGH	STRUCTURE	AUTOMATIC					315	316	317	318	319						
				320																
118	0000H	9	RIGH	STRUCTURE	BASED(PTR)				130	133										
113	0000H	9	RIGH	STRUCTURE	BASED(PTR)				114											
387	0004H	2	RIGHVAL	WORD	AUTOMATIC				394	396	397	398	399	400						
				401																
352	0004H	2	RIGHVAL	WORD	AUTOMATIC				358	360	361	362	363	364						
				365	366															
386	06DFH	208	RR	PROCEDURE	REENTRANT	STACK=0080H								417						
5			RST	LITERALLY																
5			RSS	LITERALLY																
57	0000H	2	S	WORD	PARAMETER				58											
215	0002H	1	SAVE	BYTE	AUTOMATIC				245	247										
69	0000H	2	SAVSTACK	WORD					72	446										
21	0000H		SCAN	PROCEDURE	EXTERNAL(4)	STACK=0000H								90	105					
				144	145	194	226	229	250	263	268	276	487							
7			SEGMENT	LITERALLY					143	223	237	292								
7			SEGMCOUNT	LITERALLY																
452	0004H	2	SEGMENT	WORD	MEMBER(OPER)															
445	0004H	2	SEGMENT	WORD	MEMBER(OPER)															
80	0004H	2	SEGMENT	WORD	MEMBER(OPER)															
430	0004H	2	SEGMENT	WORD	MEMBER(RIGHT)															
430	0004H	2	SEGMENT	WORD	MEMBER(LEFT)															
421	0004H	2	SEGMENT	WORD	MEMBER(RIGHT)															
421	0004H	2	SEGMENT	WORD	MEMBER(LEFT)															
409	0004H	2	SEGMENT	WORD	MEMBER(LEFT)															
387	0004H	2	SEGMENT	WORD	MEMBER(RIGHT)															
387	0004H	2	SEGMENT	WORD	MEMBER(LEFT)															
372	0004H	2	SEGMENT	WORD	MEMBER(RIGHT)															
372	0004H	2	SEGMENT	WORD	MEMBER(LEFT)															

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

352	0004H	2	SEGMENT.	WORD MEMBER(RIGHT)	
352	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
337	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
32	0004H	2	SEGMENT.	WORD MEMBER(OPERANDS)	
325	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
286	0004H	2	SEGMENT.	WORD MEMBER(RIGHT)	317
286	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	294 317
258	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
215	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	224 234
197	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
158	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
152	0004H	2	SEGMENT.	WORD MEMBER(OPER)	
118	0004H	2	SEGMENT.	WORD MEMBER(RIGHT)	
118	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
113	0004H	2	SEGMENT.	WORD MEMBER(RIGHT)	
113	0004H	2	SEGMENT.	WORD MEMBER(LEFT)	
478	0004H	2	SEGMENT.	WORD MEMBER(OPER)	
136	0145H	63	SEGBVER.	PROCEDURE BYTE STACK=0006H	326
325	0002H	1	SEGBEG.	BYTE AUTOMATIC	326 329
137	0000H	1	SEGBEG.	BYTE BASED(PT)	142 143
7			SEGTYPEBIT	LITERALLY	143 235 329
7			SEGTYPECOUNT . . .	LITERALLY	143 223 235
452	0003H	1	SFLAG.	BYTE MEMBER(OPER)	469
445	0003H	1	SFLAG.	BYTE MEMBER(OPER)	
80	0003H	1	SFLAG.	BYTE MEMBER(OPER)	
430	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
430	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
421	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
421	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
409	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
387	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
387	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
372	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
372	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
352	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
352	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
337	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
32	0003H	1	SFLAG.	BYTE MEMBER(OPERANDS)	
325	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	329
286	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	320
286	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	292 300 320
258	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
215	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	219 223 235 237
197	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
158	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	163
152	0003H	1	SFLAG.	BYTE MEMBER(OPER)	
118	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
118	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
113	0003H	1	SFLAG.	BYTE MEMBER(RIGHT)	
113	0003H	1	SFLAG.	BYTE MEMBER(LEFT)	
478	0003H	1	SFLAG.	BYTE MEMBER(OPER)	
			SHL.	BUILTIN	143 223 235 364
			SHR.	BUILTIN	366
18	0000H	2	SOURCE.	WORD PARAMETER	19
2			SPACE.	LITERALLY	
3			SPEC.	LITERALLY	210
23	0000H		SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(5) STACK=0000H	88

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

84	003EH	54	SPECHEP3ER	198 200 259 265 271 483						
324	0536H	61	SS	PROCEDURE BYTE STACK=0000H	338	354	374			
			STACKPTR	PROCEDURE REENTRANT STACK=0040H		348				
69	0007H	600	STACK	BUILTIN 72 446 447						
12	0000H	2	STRADR	BYTE ARRAY(600) 447						
9	0000H	2	STRADR	WORD PARAMETER 13						
3			STRING	WORD PARAMETER 10						
2			STRUC.	LITERALLY 232						
				LITERALLY 32 80 113 118 152 158 197						
				215 258 285 325 357 352 372 337 409 421						
				430 445 452 478						
452	0002H	1	STYPE.	BYTE MEMBER(OPER) 459						
445	0002H	1	STYPE.	BYTE MEMBER(OPER)						
80	0002H	1	STYPE.	BYTE MEMBER(OPER)						
430	0002H	1	STYPE.	BYTE MEMBER(RIGHT)						
430	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
421	0002H	1	STYPE.	BYTE MEMBER(RIGHT)						
421	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
409	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
387	0002H	1	STYPE.	BYTE MEMBER(RIGHT)						
387	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
372	0002H	1	STYPE.	BYTE MEMBER(RIGHT)						
372	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
352	0002H	1	STYPE.	BYTE MEMBER(RIGHT)						
352	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
337	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
32	0002H	1	STYPE.	BYTE MEMBER(OPERANDS)						
325	0002H	1	STYPE.	BYTE MEMBER(LEFT)						
286	0002H	1	STYPE.	BYTE MEMBER(RIGHT)	316					
286	0002H	1	STYPE.	BYTE MEMBER(LEFT)	316					
258	0002H	1	STYPE.	BYTE MEMBER(LEFT)	278					
215	0002H	1	STYPE.	BYTE MEMBER(LEFT)	218	231	233			
197	0002H	1	STYPE.	BYTE MEMBER(LEFT)	209					
158	0002H	1	STYPE.	BYTE MEMBER(LEFT)	162	170	174	187		
152	0002H	1	STYPE.	BYTE MEMBER(OPER)	154					
118	0002H	1	STYPE.	BYTE MEMBER(RIGHT)	130					
118	0002H	1	STYPE.	BYTE MEMBER(LEFT)	130	134				
118	0266H	1	STYPE.	BYTE 130 131 134						
113	0002H	1	STYPE.	BYTE MEMBER(RIGHT)	114					
113	0002H	1	STYPE.	BYTE MEMBER(LEFT)	114					
478	0002H	1	STYPE.	BYTE MEMBER(OPER)						
18	0000H	2	SYMBADR.	WORD PARAMETER 19						
15	0000H	2	SYMBADR.	WORD PARAMETER 16						
3			SYMBOL	LITERALLY						
158	0002H	2	SYMBPTR.	WORD 167 169 183 183						
196	026CH	96	SYMTYP	PROCEDURE BYTE STACK=0018H	216					
33	0000H	2	T.	WORD PARAMETER 34						
2			TAB.	LITERALLY						
36	0000H	2	TEXTADR.	WORD PARAMETER 37						
2			THENDO	LITERALLY 88 98 103 138 140 159 167						
				171 181 183 202 221 240 259 271 287 326						
				338 342 377 389 410 435 457 483 485						
23	0000H	1	FOK.	BYTE PARAMETER 24						
32	0000H	4	TOKEN.	STRUCTURE EXTERNAL(14)	98	102	138	142		
				159 162 163 164 202						
2			TRUE	LITERALLY 173 191 453 473 480 481 501						
351	05BAH	203	TT	PROCEDURE REENTRANT STACK=005CH	373	375				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

120	0004H	1	TYP.	BYTE PARAMETER AUTOMATIC	121	122	124	126
32	0000H	1	TYPE	BYTE MEMBER(TOKEN)	98	138	139	162
7			TYPEBIT.	LITERALLY 300 320				
7			TYPECOUNT.	LITERALLY				
117	0004H	73	TYPEFIND	PROCEDURE STACK=0019H	376			
120	0110H	40	TYPEEND	PROCEDURE BYTE STACK=0006H		130		
69	0262H	1	UDEFFLAG	BYTE 75 175 191 445 454 457				
3			UDEFSYM8	LITERALLY 171				
8			UDEFSYM8UL	LITERALLY 176 190				
54	0000H		UPPER.	PROCEDURE BYTE EXTERNALC(27) STACK=0000H				
286	0002H	2	VAL.	WORD AUTOMATIC 306 307 308				
32	0002H	2	VALUE.	WORD MEMBER(TOKEN) 102 142 154				
3			VARIABLE	LITERALLY 119 124 233				
4			WRD.	LITERALLY 219 223				

MODULE INFORMATION:

CODE AREA SIZE = 0966H 24380
 CONSTANT AREA SIZE = 0022H 340
 VARIABLE AREA SIZE = 026AH 6180
 MAXIMUM STACK SIZE = 006AH 1860
 801 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE BREXPR
OBJECT MODULE PLACED IN BREXPR.OBJ

COMPILER INVOKED BY: :FO: BREXPR.PLX DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

1 \$title ("BRACKET EXPRESSION")

brexpr:

/*

modified 4/13/81 R. Silberstein

*/

do;

\$include (:f1:macro.lit)

= \$nolist

\$include (:f1:brexpr.x86)

= \$nolist

\$include (:f1:ermod.ext)

= \$nolist

\$include (:f1:exglob.ext)

= \$nolist

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

$reject

/* compute index expression within brackets */

17 1  bracketexpr: proc(pt) byte public;
18 2      dcl pt address,oper based pt operandstruc,
          (firsttype,firstreg,lasttype,lastreg) byte,
          baseregi lit "0",indexregi lit "1";

19 2      regtyp: proc(pt1,pt2) byte;
20 3          dcl (pt1,pt2) address,(typ based pt1,regi based pt2) byte;
21 3          if (token.type=reg) and (token.descr=rd) then$do
22 4              typ=baseregi;
23 4              reg=token.value;
24 4              if (reg=rbp) or (reg=rbx) then return true;
25 4              typ=indexregi;
26 4              if (reg=rsi) or (reg=rax) then return true;
27 4          endif;
28 4          return false;
29 3      end regtyp;

30 2      setoperflags: proc (pt1,pt2);
31 3          dcl (pt1,pt2) address,(typ based pt1,regi based pt2) byte;
32 3          if typ=indexregi then$do
33 4              oper.sflag=oper.sflag or iregbit;
34 4              IF REGI = RDI THEN OPER.BASEINDEX = OPER.BASEINDEX OR INDEXREGBIT;
35 4          else$do
36 4              oper.sflag=oper.sflag or bregbit;
37 4              IF REGI = RBP THEN OPER.BASEINDEX = OPER.BASEINDEX OR BASEREGBIT;
38 4          endif;
39 3      end setoperflags;

40 2      if not regtyp(.firsttype,.firstreg) then return false;
41 2      call setoperflags(.firsttype,.firstreg);
42 2      call scan;
43 2      if specialtoken('+') then$do
44 3          call scan;
45 3          if not regtyp(.lasttype,.lastreg) then return false;
46 3          if firsttype=lasttype then return false;
47 3          call setoperflags(.lasttype,.lastreg);
48 3          call scan;
49 3      endif;
50 2      if not specialtoken(rightbracket) then return false;
51 2      call scan;
52 2      if (oper.sflag and segmbit) = 0 then$do
53 3          if oper.stype=number then$do
54 4              oper.baseindex=oper.baseindex or nooverridebit;
55 4          endif;
56 3      endif;
57 2      oper.stype=variable;
58 2      return true;
59 2      end bracketexpr;

60 1      end$module bexpr;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

16	0000H	1	ACCLN	BYTE EXTERNAL(11)	
16	0000H	80	ACCU	BYTE ARRAY(80) EXTERNAL(12)	
2			ADDR	LITERALLY	16 18
16	0008H	1	BASEINDEX	BYTE MEMBER(OPERANDS)	
18	0008H	1	BASEINDEX	BYTE MEMBER(OPER)	39 44 68
7			BASEREGBIT	LITERALLY	44
7			BASEREGCOUNT	LITERALLY	
18			BASEREGT	LITERALLY	23
17	0000H	148	BRACKETEXPR	PROCEDURE BYTE PUBLIC STACK=000AH	
7			BREGBIT	LITERALLY	42
7			BREGCOUNT	LITERALLY	
1	0000H		BREXPR	PROCEDURE STACK=0000H	
4			BYT	LITERALLY	
16	0000H	2	CTP	WORD EXTERNAL(3)	
3			CODE	LITERALLY	
2			CR	LITERALLY	
16	0000H	1	CSEGSPEC	BYTE EXTERNAL(5)	
16	0000H	1	CSEGTTYPE	BYTE EXTERNAL(4)	
16	0000H	2	CSEGVALUE	WORD EXTERNAL(5)	
16	0000H	2	CURDSEG	WORD EXTERNAL(8)	
2			DCL	LITERALLY	
3			DELETEDSYM	LITERALLY	
16	0001H	1	DESCR	BYTE MEMBER(TOKEN)	21
3			DOUBLEDEFINED	LITERALLY	
16	0000H	1	DSPEC	BYTE EXTERNAL(7)	
4			DWRD	LITERALLY	
2			ELSEDD	LITERALLY	39
2			ENDCASE	LITERALLY	
2			ENDDD	LITERALLY	
2			ENDDFOREVER	LITERALLY	
2			ENDIF	LITERALLY	29 44 69
2			ENDMODULE	LITERALLY	
2			ENDOFFILE	LITERALLY	
2			ENDPROC	LITERALLY	
2			ENDWHTLE	LITERALLY	
13	0000H		ERRMSG	PROCEDURE EXTERNAL(2) STACK=0000H	
13	0000H	1	ERRNO	BYTE PARAMETER	14
2			FALSE	LITERALLY	31 48 55 57 62
18	0003H	1	FIRSTREG	BYTE	47 49
18	0002H	1	FIRSTTYPE	BYTE	47 49 56
2			FOREVER	LITERALLY	
2			FORMFEED	LITERALLY	
3			IDENT	LITERALLY	
7			INDEXREGBIT	LITERALLY	39
7			INDEXREGCOUNT	LITERALLY	
18			INDEXREGT	LITERALLY	27 35
2			INIT	LITERALLY	
7			IREGBIT	LITERALLY	37
7			IREGCOUNT	LITERALLY	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

3		LAB.	LITERALLY																	
18	0005H	1	LASTREG.	BYTE	54	58														
18	0004H	1	LASTTYPE	BYTE	54	56	58													
6			LEFTBRACKET.	LITERALLY																
16	0000H	2	LENGTH	WORD MEMBER(COPERANDS)																
18	0000H	2	LENGTH	WORD MEMBER(COPER)																
2			LF	LITERALLY																
2			LIT.	LITERALLY		2	3	4	5	6	7	14								
3			NEGLECTED.	LITERALLY																
16	0000H	1	NEXTCH	BYTE EXTERNAL(10)																
4			NIL.	LITERALLY																
16	0000H	1	NOOPER	BYTE EXTERNAL(13)																
7			NOOVERCOUNT.	LITERALLY																
7			NOOVERRIDEBIT.	LITERALLY		68														
3			NUMBER	LITERALLY		66														
6			OAND	LITERALLY																
6			OEQ.	LITERALLY																
16	0006H	2	OFFSET	WORD MEMBER(COPERANDS)																
18	0006H	2	OFFSET	WORD MEMBER(COPER)																
6			OGE.	LITERALLY																
6			OGT.	LITERALLY																
6			OLAST.	LITERALLY																
6			OLE.	LITERALLY																
6			OLENGTH.	LITERALLY																
6			OLT.	LITERALLY																
6			OMOD	LITERALLY																
6			ONE.	LITERALLY																
6			ONOT	LITERALLY																
6			OOFFSET.	LITERALLY																
6			OOR.	LITERALLY																
18	0000H	9	OPER	STRUCTURE BASED(PT)		37	39	42	44	64										
				66 68 71																

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

2		REENT.	LITERALLY				
3		REG.	LITERALLY	21			
20	0000H	1 REGT.	BYTE BASED(PT2)		24	25	28
34	0000H	1 REGT.	BYTE BASED(PT2)		33	43	
19	0094H	73 REGTYP.	PROCEDURE BYTE STACK=0006H				47 54
5		RES.	LITERALLY				
6		RIGHTBRACKET.	LITERALLY	61			
5		RSL.	LITERALLY	28			
5		RSS.	LITERALLY				
8	0000H	SCAN.	PROCEDURE EXTERNAL(0) STACK=0000H				50 53
			59 63				
7		SEGMPIT.	LITERALLY	64			
7		SEGMCOUNT.	LITERALLY				
18	0004H	2 SEGMENT.	WORD MEMBER(OPER)				
16	0004H	2 SEGMENT.	WORD MEMBER(OPERANDS)				
7		SEGTYPEBIT.	LITERALLY				
7		SEGTYPECOUNT.	LITERALLY				
33	000DH	65 SETOPERFLAGS.	PROCEDURE STACK=0006H				49 58
18	0003H	1 SFLAG.	BYTE MEMBER(OPER)		37	42	64
16	0003H	1 SFLAG.	BYTE MEMBER(OPERANDS)				
2		SPACE.	LITERALLY				
3		SPEC.	LITERALLY				
10	0000H	SPECIALTOKEN.	PROCEDURE BYTE EXTERNAL(1) STACK=0000H				51
			61				
3		STRING.	LITERALLY				
2		STRUC.	LITERALLY	16	16		
18	0002H	1 STYPE.	BYTE MEMBER(OPER)		66	71	
16	0002H	1 STYPE.	BYTE MEMBER(OPERANDS)				
3		SYMBOL.	LITERALLY				
2		TAB.	LITERALLY				
2		THENDO.	LITERALLY	21	35	51	64 66
10	0000H	1 TOK.	BYTE PARAMETER	11			
16	0000H	4 TOKEN.	STRUCTURE EXTERNAL(9)				21 24
2		TRUE.	LITERALLY	26	29	72	
34	0000H	1 TYP.	BYTE BASED(PT1)		35		
20	0000H	1 TYP.	BYTE BASED(PT1)		23	27	
16	0000H	1 TYPE.	BYTE MEMBER(TOKEN)			21	
7		TYPEBIT.	LITERALLY				
7		TYPECOUNT.	LITERALLY				
3		UDFSYMP.	LITERALLY				
16	0002H	2 VALUE.	WORD MEMBER(TOKEN)		24		
3		VARIABLE.	LITERALLY	71			
4		WRD.	LITERALLY	21			

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

MODULE INFORMATION:

CODE AREA SIZE	= 011EH	2860
CONSTANT AREA SIZE	= 0000H	00
VARIABLE AREA SIZE	= 0006H	60
MAXIMUM STACK SIZE	= 000AH	100
254 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION



ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE PSEUDOM

OBJECT MODULE PLACED IN PSEUD1.OBJ

COMPILER INVOKED BY: :F0: PSEUD1.PLK DEBUG OPTIMIZED(2) DATE(10/5/81) PAGERWIDTH(100) YR81

1 \$title ("PSEUDO INSTRUCTION MODULE-1")

pseudom:

do;

/*

modified 4/9/81 R. Silberstein

modified 4/15/81 R. Silberstein

modified 5/7/81 R. Silberstein

modified 7/24/81 R. Silberstein

modified 8/26/81 R. Silberstein

modified 8/19/81 R. Silberstein

*/

/*

This is the module to perform the decoding of
all legal pseudo instructions of the assembler.
There is one subroutine for each corresponding
pseudoinstruction.

*/

```

= $include (:f1:macro.lit)
= $nolist
= $include (:f1:struc.lit)
= $nolist
= $include (:f1:equals.lit)
= $nolist
= $include (:f1:pseud1.x86)
= $nolist
= $include (:f1:outp.lit)
= $nolist
= $include (:f1:subr2.ext)
= $nolist
= $include (:f1:print.ext)
= $nolist
= $include (:f1:scan.ext)
= $nolist
= $include (:f1:symb.ext)
= $nolist
= $include (:f1:expr.ext)
= $nolist
= $include (:f1:ermod.ext)
= $nolist
= $include (:f1:outp.ext)
= $nolist
= $include (:f1:global.ext)
= $nolist

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

reject
/***** COMMAND SUBROUTINES *****/

/* routine to test if rest of line is either a comment or empty -
   if not, print error message - skip rest of line */

```

```

130 1 test$emptyline: proc;
131 2     if not emptyline then call errmsg(end$of$line$err);
133 2     call skip$rest$of$line;
134 2 end test$emptyline;

```

```

/* list current address in front of printline */

```

```

135 1 list$ip: proc PUBLIC;
136 2     if (prefixptr=0) and (pass <> 0) then$do
138 3         call hex2out(cip,.prefix(1));
139 3         prefixptr=6;
140 3     end$if;
141 2 end list$ip;

```

```

/* common routine for ORG and RS (reserve storage pseudo) */

```

```

142 1 org$rs: proc (disp,typ);
143 2     dcl disp addr,typ byte,oper operandstruc at (.operands(0));

144 2     if noforwardexpr(.oper) then$do /* evaluate operand */
146 3         if oper.type=number then$do
148 4             current$symbol.length=oper.offset;
149 4             cip=disp+oper.offset*typ; /* compute new instruction pointer */
150 4             call test$emptyline;
151 4             return;
152 4         end$if;
152 4     end$if;

```

```

/* error in expression */

```

```

154 2     call errmsg(pseudo$operr);
155 2     call skip$rest$of$line;
156 2 end org$rs;

```

```

/* perform handling for CSEG,DSEG,SSEG,ESSEG routines */

```

```

157 1 segment$trout: proc (p1,p2,p3,segr);
158 2     dcl segr byte,(p1,p2,p3) address,
        current$seg based p1 addr,
        seg$specified based p2 byte,
        cip$save based p3 addr,
        oper operandstruc at (.operands(0)),
        low byte at (.csegvalue),high byte at (.csegvalue+1);

```

```

159 2 emit: proc;
160 3     dcl datatab(4) byte data (ESvalue,CSvalue,SSvalue,DSvalue);
161 3     call emit$code$byte(high,datatab(segr));
162 3     call emit$code$byte(low,datatab(segr));
163 3     call hex2out(csegvalue,.prefix(3)); /* print value on print line */
164 3     prefixptr=7;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

165 3      end emit;

166 2      do case csegtype; /* save current segment attributes */
167 3          do; cureseg=csegvalue; espec=csegspec; escip=cip; end; /* IS */
172 3          do; curcseg=csegvalue; cspec=csegspec; cscip=cip; end; /* CS */
177 3          do; cursseg=csegvalue; sspec=csegspec; scip=cip; end; /* SS */
182 3          do; curdseg=csegvalue; dspec=csegspec; dscip=cip; end; /* DS */
187 3      endcase;

188 2      if emptyline then$do /* allow no parameter */
190 3          call skip$rest$of$line;
191 3          csegvalue=0;
192 3          csegtype=segr;
193 3          csegspec=false; /* no segment value specified */
194 3          cip=0;
195 3          return;
196 3      end$if;

197 2      if specialtoken('$') then$do /* allow "$" */
199 3          csegtype=segr; /* pick up previous values */
200 3          csegspec=segspecified;
201 3          csegvalue=currentseg;
202 3          cip=cipsave;
203 3          if csegspec then call emit;
205 3          call scan; /* skip $ */
206 3          call test$emptyline;
207 3          return;
208 3      end$if;

209 2      if expression(.oper) then$do /* operand must be expression */
211 3          if oper.stype=number then$do
213 4              csegvalue=oper.offset; /* pick up segment value */
214 4              csegtype=segr;
215 4              csegspec=true; /* value is specified */
216 4              cip=0;
217 4              call emit;
218 4              call test$emptyline;
219 4              return;
220 4          end$if;
220 4      end$if;

/* must be illegal operand */
222 2      call skip$rest$of$line;
223 2      call errmsg(pseudooperr);

224 2      end segmentrout;

/* common routine for DB,DW and DD */
225 1      DB$DW$DD$common: proc(n);
226 2          dcl(n,continue) byte,lg addr;
227 2          DECLARE EP BYTE;

228 2      item: proc(n); /* find one element of element list */
229 3          dcl (n,i,errorprinted) byte,

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

        oper operandstruct at (.oper.offset),
        low   byte at (.oper.offset),
        high  byte at (.oper.offset+1),
        seglow byte at (.oper.segment),
        seghigh byte at (.oper.segment+1);
230  3      emit: proc (outputbyte);
231  4          ucl outputbyte byte,
                datatab(4) byte data (ESdata,CSdata,SSdata,DSdata);
232  4          call emitcodebyte(outputbyte,datatab(csectype));
233  4      end emit;

234  3      locexpr: proc byte;
235  4          if expression(.oper) then$do
237  5              i=.oper.stype;
238  5              if (i=number) or (i=variable) or (i=lab) then return true;
                end$if;
241  4          return false;
242  4      end locexpr;

243  3      DBhandle: proc;
244  4          if (token.type=string) and (acclen > 1) then$do
246  5              lg=lg+acclen-1;
247  5              i=0ffh;
248  5              do while (i:=i+1) < acclen;
249  6                  call emit(accum(i));
250  6              end$while;
251  5              oper.stype=number; /* dummy */
252  5              call scan; /* skip string */
253  5          else$do
255  5              if locexpr then$do
257  6                  call emit(low);
258  6              else$do
260  6                  call emit(0);
261  6                  call errmsg(illexprelem);
262  6              end$if;
263  5          end$if;
264  4      end DBhandle;

265  3      DWhandle: proc;
266  4          if locexpr then$do
268  5              call emit(low);
269  5              call emit(high);
270  5          else$do
272  5              call emit(0);
273  5              call emit(0);
274  5              call errmsg(illexprelem);
275  5          end$if;
276  4      end DWhandle;

277  3      DDhandle: proc;
278  4          if locexpr then$do
280  5              if oper.stype <> number then$do
282  6                  if (oper.sflag and segmbit) <> 0 then$do
284  7                      call emit(low);
285  7                      call emit(high);
286  7                      call emit(seglow);
287  7                      call emit(seghigh);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

288 7      return;
289 7      else$do
291 7          call errmsg(missegmint0);
292 7      end$if;
293 6      end$if;
        end$if;
295 4      do i=0 to 3; call emit(0); end$do; /* dummy */
298 4      call errmsg(illexprelem);
299 4      end DDhandle;

        /* ITEM main program */
300 3      lg=lg+1;
301 3      do case n;
302 4          call DBhandle;
303 4          call DWhandle;
304 4          call DDhandle;
305 4      end$case;
306 3      if specialtoken(",") then$do
308 4          call scan;
309 4          continue=true;
310 4      else$do
312 4          if emptyline then$do
314 5              call skiprest$of$line;
315 5          else$do
317 5              CALL ERRMSG (ENDOFLINEERR);
318 5              CALL SKIPRESTIOFLINE;
319 5          end$if;
320 4          end$if;
321 3      end item;

        /* DB$DW$DD$common main program */
322 2      CALL LISTCIP;
323 2      EP = FALSE;
324 2      lg=0;
325 2      continue=true;
326 2      do while continue;
327 3          errorprinted=false;
328 3          continue=false;
329 3          call item(n);
330 3          EP = EP OR ERRORPRINTED;
331 3      end$while;
332 2      currentsymbol.length=lg;
333 2      ERRORPRINTED = EP; /* SD SOURCE LINE IS ECHOED IF ERROR */
334 2      end DB$DW$DD$common;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

reject

/***** PSEUDO SUBROUTINES *****/

```

335 1  DRrout: proc public;
336 2      call DB$DW$DD$common(0);
337 2  end DRrout;

338 1  DWrout: proc public;
339 2      call DB$DW$DD$common(1);
340 2  end DWrout;

341 1  DDrout: proc public;
342 2      call DB$DW$DD$common(2);
343 2  end DDrout;

344 1  RSrout: proc (typ) public;
345 2      dcl typ byte;
346 2      call listcip; /* list current address on printline */
347 2      call orgrs(cip,typ); /* cip = cip + typ * expression */
348 2  end RSrout;

349 1  CSEGrout: proc public;
350 2      call segmentrout(.curcseg,.cspec,.cscip,rcs);
351 2  end CSEGrout;

352 1  DSEGrout: proc public;
353 2      call segmentrout(.curdseg,.dspec,.dscip,rds);
354 2  end DSEGrout;

355 1  SSEGrout: proc public;
356 2      call segmentrout(.cursseg,.sspec,.sscip,rss);
357 2  end SSEGrout;

358 1  ESEGrout: proc public;
359 2      call segmentrout(.cureseg,.espec,.escip,res);
360 2  end ESEGrout;

361 1  ORGrout: proc public;
362 2      call orgrs(0,byt); /* cip = 0 + expression */
363 2  end ORGrout;

364 1  EQUrout: proc public;
365 2      dcl oper operandstruc at (.operands(0)),
366 2          macdefpt based codemacroptr address;
367 3      codempossible: proc byte;
368 3          return (nextch=cr or nextch=';');
369 3      end codempossible;

369 2      do case pass;
370 3          do; /* pass 0 */
371 4              if codempossible and
372 4                  findcodemacro(acclen,.accum(0),.codemacroptr) then$do
373 5                  currentsymbol.stype=code;
374 5                  call enterattributes(symtabadr,.currentsymbol);
375 5                  if not newmacro(acclensave,.accumsave,macdefpt) then

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

376 5          fullsymtab=true;
      elseifdo
379 5          nooper=0; /* find normal operand expression */
380 5          IF NOFORWARDOPER THEN$DO
382 6            call enterattributes(symtabadr,.operands(0));
383 6            call skip$rest$of$line;
384 6          elseifdo
386 6            currentsymbol.stype=undefsymb;
387 6            call enterattributes(symtabadr,.currentsymbol);
388 6            call skip$rest$of$line;
389 6          endif;
390 5        endif;
391 4      end;

392 3      do; /* pass 1 */
393 4      if currentsymbol.stype <> code then$do /* update symbol value */
395 5        nooper=0;
396 5        IF NOFORWARDOPER THEN$DO
398 6          call enterattributes(symtabadr,.operands(0));
399 6          endif;
401 4        endif;
402 4        call skip$rest$of$line;
402 4      end;

403 3      do; /* pass 2 - scan to produce possible error messages */
404 4      if currentsymbol.stype=code then$do
406 5        call scan;
407 5      elseifdo
409 5        nooper=0;
410 5        IF NOT NOFORWARDOPER OR (CURRENTSYMBOL.STYPE = ERROR) THEN$DO
412 6          call errmsg(pseudooperr);
413 6          call skip$rest$of$line; /* only one error message */
414 6        elseifdo
416 6          prefixptr=7;
417 6          call hex2out(oper.offset,.prefix(3));
418 6          endif;
419 5        endif;
420 4        call test$emptyline;
421 4      end;

422 3      end$case;
423 2    end EQUout;

424 1  end$module pseudom;

```

CCP/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
127	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(77)
127	0000H	1	ACCLEN. BYTE EXTERNAL(81) 244 246 248 371
127	0000H	1	ACCLENSAVE. BYTE EXTERNAL(84) 375
127	0000H	80	ACCUM. BYTE ARRAY(80) EXTERNAL(82) 249 371
127	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(83) 375
2			ADDR. LITERALLY 12 18 36 39 42 45 77 78 84 97 90 93 96 99 102 105 127 143 158 225 229 365
26	0000H		ALPHANUMERIC. PROCEDURE BYTE EXTERNAL(5) STACK=0000H
29	0000H		ASCICHAR. PROCEDURE BYTE EXTERNAL(6) STACK=0000H
104	0000H	2	ASCIIPTR. WORD PARAMETER 105
229	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
365	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
127	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
158	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
127	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
143	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
3			BASEREGBIT. LITERALLY
3			BASEREGCOUNT. LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
5			BYT. LITERALLY 362
20	0000H	1	CH. BYTE PARAMETER 21
29	0000H	1	CH. BYTE PARAMETER 30
124	0000H	1	CH. BYTE PARAMETER 125
47	0000H	1	CH. BYTE PARAMETER 48
26	0000H	1	CH. BYTE PARAMETER 27
23	0000H	1	CH. BYTE PARAMETER 24
17	0000H	1	CH. BYTE PARAMETER 18
32	0000H	1	CH. BYTE PARAMETER 33
127	0000H	2	CIP. WORD EXTERNAL(46) 138 149 170 175 180 185 194 202 216 347
158	0000H	2	CTPSAVE. WORD BASED(P3) 202
4			CODE. LITERALLY 373 393 404
127	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(103)
3			CODEMACROHEAD. LITERALLY
127	0000H	2	CODEMACROPTR. WORD EXTERNAL(98) 365 371 375
366	0511H	29	CODEMPOSSIBLE. PROCEDURE BYTE STACK=0004H 371
226	000AH	1	CONTINUE. BYTE 309 325 326 328
2			CR. LITERALLY 367
127	0000H	2	CSCPT. WORD EXTERNAL(51) 175 350
10			CSDATA. LITERALLY 231
349	03BCH	23	CSEGR0UT. PROCEDURE PUBLIC STACK=0010H
127	0000H	1	CSEGSPEC. BYTE EXTERNAL(49) 169 174 179 184 193 200 203 215
127	0000H	1	CSEGTTYPE. BYTE EXTERNAL(47) 166 192 199 214 232
127	0000H	2	CSEGVAlUE. WORD EXTERNAL(48) 158 163 168 173 178 183 191 201 213
127	0000H	1	CSPEC. BYTE EXTERNAL(58) 174 350

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

16		CSVALUE.	LITERALLY	156			
127	0000H	2 CURCSEG.	WORD EXTERNAL(54)	173	356		
127	0000H	2 CURDSEG.	WORD EXTERNAL(55)	173	356		
127	0000H	2 CURSEGC.	WORD EXTERNAL(57)	168	359		
158	0000H	2 CURRENTSEG.	WORD BASED(P1)	261			
127	0000H	9 CURRENTOH.	STRUCTURE EXTERNAL(94)	148	332	373	374
			386 387 393 404 410				
127	0000H	2 CURSSEG.	WORD EXTERNAL(55)	176	356		
35	0000H	2 D.	WORD PARAMETER	36			
41	0000H	2 D.	WORD PARAMETER	42			
44	0000H	2 D.	WORD PARAMETER	45			
38	0000H	2 D.	WORD PARAMETER	39			
231	0004H	4 DATATAB.	BYTE ARRAY(4) DATA	232			
160	0000H	4 DATATAB.	BYTE ARRAY(4) DATA	161	162		
225	01BFH	70 DBOWDDCOMMON.	PROCEDURE STACK=001AH	336	339	342	
243	0294H	105 DBHANDLE.	PROCEDURE STACK=000EH	302			
335	0387H	11 DBROUT.	PROCEDURE PUBLIC STACK=001EH				
2		DCL.	LITERALLY				
277	032BH	92 DBHANDLE.	PROCEDURE STACK=000EH	304			
341	039DH	11 DBROUT.	PROCEDURE PUBLIC STACK=001EH				
44	0000H	DECOU.	PROCEDURE EXTERNAL(11) STACK=000CH				
127	0000H	1 DEFAULTDRTVE.	BYTE EXTERNAL(101)				
4		DELETEDSYMB.	LITERALLY				
127	0001H	1 DESCR.	BYTE MEMBER(TOKEN)				
92	0000H	2 DEST.	WORD PARAMETER	93			
20	0000H	DTGTT.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H				
142	0006H	2 DISP.	WORD PARAMETER AUTOMATIC	143	149		
4		DOUBLEDEFINED.	LITERALLY				
10		DRCODEDATA.	LITERALLY				
10		DRCODESEGM.	LITERALLY				
10		DRDATAATA.	LITERALLY				
10		DRDATASEGM.	LITERALLY				
10		DREXTRADATA.	LITERALLY				
10		DREXTRASEGM.	LITERALLY				
10		DRSTACKDATA.	LITERALLY				
10		DRSTACKSEGM.	LITERALLY				
127	0000H	2 DSCTP.	WORD EXTERNAL(53)	185	353		
10		DSDATA.	LITERALLY	231			
352	03D3H	23 DSEGROUT.	PROCEDURE PUBLIC STACK=0010H				
127	0000H	1 DSPEC.	BYTE EXTERNAL(59)	184	353		
10		DSVALUE.	LITERALLY	160			
265	02FDH	46 DWHANDLE.	PROCEDURE STACK=000EH	303			
5		DWRD.	LITERALLY				
338	0392H	11 DWROUT.	PROCEDURE PUBLIC STACK=001EH				
61	0000H	EJECT.	PROCEDURE EXTERNAL(18) STACK=0000H				
2		ELSECO.	LITERALLY	376			
159	0188H	55 EMIT.	PROCEDURE STACK=0000H	204	217		
230	0255H	23 EMIT.	PROCEDURE STACK=0004H	249	257	260	263
			269 272 273 284 285 286 287	296			
124	0000H	EMITCODEBYTE.	PROCEDURE EXTERNAL(44) STACK=0000H	161	162		
			232				
120	0000H	EMITINIT.	PROCEDURE EXTERNAL(42) STACK=0000H				
122	0000H	EMITTERMINATE.	PROCEDURE EXTERNAL(43) STACK=0000H				
75	0000H	EMPTYLINE.	PROCEDURE BYTE EXTERNAL(24) STACK=0000H	131			
			188 312				
2		ENDCASE.	LITERALLY				
2		ENODO.	LITERALLY				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		ENDFOREVER	LITERALLY		
2		ENDJF	LITERALLY	152 220 239 293 309	
2		ENDMODULE	LITERALLY		
2		ENDOFFILL	LITERALLY		
9		ENDOFFLINEERR	LITERALLY	132 317	
78	0000H	2	ENDOSYMBTAB	WORD EXTERNAL(26)	
2			LITERALLY		
2			LITERALLY		
95	0000H		ENTERATTRIBUTES	PROCEDURE EXTERNAL(35) STACK=0000H	374 382
				387 398	
127	0000H	1	EOFSCT	BYTE EXTERNAL(85)	
10			EOFTYPE	LITERALLY	
227	0000H	1	EP	BYTE 322 330 333	
35	0000H		EQUAL	PROCEDURE BYTE EXTERNAL(8) STACK=0000H	
364	0427H	234	EQUROUT	PROCEDURE PUBLIC STACK=0000H	
117	0000H		ERRMSG	PROCEDURE EXTERNAL(41) STACK=0000H	132 154
				223 261 274 291 298 317 412	
117	0000H	1	ERRNO	BYTE PARAMETER 118	
4			ERROR	LITERALLY 410	
127	0000H	1	ERRORPRINTED	BYTE EXTERNAL(88)	327 350 333
229	0000H	1	ERRORPRINTED	BYTE	
127	0000H	2	ERRORS	WORD EXTERNAL(78)	
127	0000H	2	ESCP	WORD EXTERNAL(50)	170 359
10			ESDATA	LITERALLY 231	
358	0401H	23	ESEGR0UT	PROCEDURE PUBLIC STACK=0010H	
127	0000H	1	ESPEC	BYTE EXTERNAL(61)	169 359
10			ESVALUE	LITERALLY 160	
111	0000H		EXPRESSION	PROCEDURE BYTE EXTERNAL(39) STACK=0000H	299
				235	
2			FALSE	LITERALLY 193 241 323 327 328	
14	0000H	2	FCBADR	WORD PARAMETER 15	
14	0000H		FILEABORT	PROCEDURE EXTERNAL(1) STACK=0000H	
3			FILE1STRUCTURE	LITERALLY	
3			FILE0STRUCTURE	LITERALLY	
17	0000H		FILL	PROCEDURE EXTERNAL(2) STACK=0000H	
98	0000H		FINDCODEMACRO	PROCEDURE BYTE EXTERNAL(34) STACK=0000H	371
89	0000H		FINDSYMBOL	PROCEDURE BYTE EXTERNAL(31) STACK=0000H	
127	0000H	2	FIXSIMACROPTK	WORD EXTERNAL(89)	
2			FOREVER	LITERALLY	
2			FORMFEED	LITERALLY	
77	0000H	2	FREET	WORD EXTERNAL(25)	
127	0000H	1	FULLSYMBTAB	BYTE EXTERNAL(91)	376
92	0000H		GETATTRIBUTES	PROCEDURE EXTERNAL(32) STACK=0000H	
128	0000H		GLOBALINIT	PROCEDURE EXTERNAL(104) STACK=0000H	
127	0000H	5	HELP	BYTE ARRAY(5) EXTERNAL(99)	
38	0000H		HEX1OUT	PROCEDURE EXTERNAL(9) STACK=0000H	
41	0000H		HEX2OUT	PROCEDURE EXTERNAL(10) STACK=0000H	136 163
				417	
158	0001H	1	HIGH	BYTE EXTERNAL(48) AT 161	
229	0007H	1	HIGH	BYTE EXTERNAL(97) AT 269 285	
127	0000H	1	I	BYTE EXTERNAL(100)	
229	000CH	1	I	BYTE 237 238 247 248 249 295	
4			IDENT	LITERALLY	
127	0000H	1	IFLEVEL	BYTE EXTERNAL(93)	
127	0000H	1	IFLIST	BYTE EXTERNAL(64)	
9			ILLEXPREM	LITERALLY 261 274 298	
127	0000H	1	INCLUDEDEFAULT	BYTE EXTERNAL(102)	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

127	0000H	1	INCLUDEDN.	BYTE EXTERNAL(92)				
3			INDEXREGBIT.	LITERALLY				
3			INDEXREGCOUNT.	LITERALLY				
4			INIT.	LITERALLY				
10			INTELDATA.	LITERALLY				
127	0000H	1	INTELHEXON.	BYTE EXTERNAL(86)				
10			INTELSegment.	LITERALLY				
3			IREGBIT.	LITERALLY				
3			IREGCOUNT.	LITERALLY				
228	0205H	80	ITEM.	PROCEDURE STACK=0014H	329			
4			LAB.	LITERALLY	238			
8			LEFTBRACKET.	LITERALLY				
365	0000H	2	LENGTH.	WORD MEMBER(COPER)				
127	0000H	2	LENGTH.	WORD MEMBER(COPERANDS)				
158	0000H	2	LENGTH.	WORD MEMBER(COPER)				
143	0000H	2	LENGTH.	WORD MEMBER(COPER)				
229	0000H	2	LENGTH.	WORD MEMBER(COPER)				
127	0000H	2	LENGTH.	WORD MEMBER(CURRENTSYMBOL)	148	332		
23	0000H		LETTER.	PROCEDURE BYTE EXTERNAL(4) STACK=0000H				
2			LF.	LITERALLY				
226	0008H	2	LG.	WORD	246	300	324	332
104	0000H	1	LG.	BYTE PARAMETER	105			
101	0000H	1	LG.	BYTE PARAMETER	102			
98	0000H	1	LG.	BYTE PARAMETER	99			
89	0000H	1	LG.	BYTE PARAMETER	90			
86	0000H	1	LG.	BYTE PARAMETER	87			
83	0000H	1	LG.	BYTE PARAMETER	84			
135	0017H	35	LISTCIP.	PROCEDURE PUBLIC STACK=0008H	322	346		
2			LIT.	LITERALLY	2	3	4	5
					9	10		
234	026CH	40	LOCEXPR.	PROCEDURE BYTE STACK=0006H	255	266	278	
229	0006H	1	LOW.	BYTE EXTERNAL(97) AT	257	268	284	
158	0000H	1	LOW.	BYTE EXTERNAL(49) AT	162			
365	0000H	2	MACDEFPT.	WORD BASED(CODEMACROPTR)	375			
86	0000H	2	MACDEFPT.	WORD PARAMETER	87			
127	0000H	2	MACROPTR.	WORD EXTERNAL(90)				
127	0000H	1	MAXCOL.	BYTE EXTERNAL(65)				
9			MISSESGMINFO.	LITERALLY	291			
41	0000H	2	N.	WORD PARAMETER	42			
38	0000H	1	N.	BYTE PARAMETER	39			
35	0000H	1	N.	BYTE PARAMETER	36			
17	0000H	1	N.	BYTE PARAMETER	18			
228	0004H	1	N.	BYTE PARAMETER AUTOMATIC	229	301		
225	0004H	1	N.	BYTE PARAMETER AUTOMATIC	226	329		
44	0000H	2	N.	WORD PARAMETER	45			
4			NEGLECTED.	LITERALLY				
101	0000H		NEWCMRODY.	PROCEDURE BYTE EXTERNAL(35) STACK=0000H				
104	0000H		NEWCNAME.	PROCEDURE BYTE EXTERNAL(36) STACK=0000H				
86	0000H		NEWMACRO.	PROCEDURE BYTE EXTERNAL(30) STACK=0000H				375
83	0000H		NEWSYMBOL.	PROCEDURE BYTE EXTERNAL(29) STACK=0000H				
127	0000H	1	NEXTCH.	BYTE EXTERNAL(80)	367			
5			NIL.	LITERALLY				
127	0000H	1	NOERROR.	BYTE EXTERNAL(87)				
114	0000H		NOFORWARDEXPR.	PROCEDURE BYTE EXTERNAL(40) STACK=0000H				144
109	0000H		NOFORWARDPER.	PROCEDURE BYTE EXTERNAL(38) STACK=0000H				380
					396	410		
127	0000H	1	NOOPER.	BYTE EXTERNAL(96)	379	395	409	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

3		NOOVERRIDEIT.	LITERALLY					
3		NOOVERRIDEIT.	LITERALLY					
4		NUMBER	LITERALLY	143	211	238	251	248
8		OAND	LITERALLY					
8		OEQ.	LITERALLY					
365	0006H	2 OFFSET	WORD MEMBER(OPER)			417		
158	0006H	2 OFFSET	WORD MEMBER(OPER)			213		
143	0006H	2 OFFSET	WORD MEMBER(OPER)			148	149	
127	0006H	2 OFFSET	WORD MEMBER(OPERANDS)					
127	0006H	2 OFFSET	WORD MEMBER(CURRENTSYMBOL)					
229	0006H	2 OFFSET	WORD MEMBER(OPER)			229		
8		OGE.	LITERALLY					
8		UGT.	LITERALLY					
8		OLAST.	LITERALLY					
8		OLE.	LITERALLY					
8		OLENGTH.	LITERALLY					
8		OLT.	LITERALLY					
8		OMOD	LITERALLY					
8		ONE.	LITERALLY					
8		ONUT	LITERALLY					
8		OOFFSET.	LITERALLY					
8		OOR.	LITERALLY					
365	0000H	9 OPER	STRUCTURE EXTERNAL(97) AT			417		
229	0000H	9 OPER	STRUCTURE EXTERNAL(97) AT			229	235	237 251
			280 282					
158	0000H	9 OPER	STRUCTURE EXTERNAL(97) AT			209	211	213
143	0000H	9 OPER	STRUCTURE EXTERNAL(97) AT			144	146	148 149
107	0000H	OPERAND.	PROCEDURE BYTE EXTERNAL(37) STACK=0000H					
127	0000H	36 OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(97)				143	158
			229 365 382 398					
3		OPERANDSTRUC	LITERALLY	127	143	158	229	365
4		OPERATOR	LITERALLY					
8		OPTR	LITERALLY					
361	0418H	15 ORGROUT.	PROCEDURE PUBLIC STACK=0012H					
142	003AH	60 ORGRS.	PROCEDURE STACK=000EH			347	352	
8		OSEG	LITERALLY					
8		OSHL	LITERALLY					
8		OSHRT	LITERALLY					
8		OSHR	LITERALLY					
8		OTYPE.	LITERALLY					
230	0004H	1 OUTPUTBYTE	BYTE PARAMETER AUTOMATIC			231	232	
11	0000H	OUTTEXT.	PROCEDURE EXTERNAL(0) STACK=0000H					
8		OXOR	LITERALLY					
157	0006H	2 P1	WORD PARAMETER			158	201	
157	0004H	2 P2	WORD PARAMETER			158	200	
157	0002H	2 P3	WORD PARAMETER			158	202	
127	0000H	1 PAGESIZE	BYTE EXTERNAL(71)					
127	0000H	1 PASS	BYTE EXTERNAL(45)			136	369	
7		PCODEMACRO	LITERALLY					
7		PCSEG.	LITERALLY					
7		PDB.	LITERALLY					
7		PDBTT.	LITERALLY					
7		PDD.	LITERALLY					
7		PDSEG.	LITERALLY					
7		PDW.	LITERALLY					
7		PEJECT	LITERALLY					
7		PEND	LITERALLY					

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

7		PENDIF	LITERALLY										
7		PEND?	LITERALLY										
7		PEQU	LITERALLY										
7		PESEG	LITERALLY										
7		PIF	LITERALLY										
7		PIFLIST	LITERALLY										
7		PTINCLUDE	LITERALLY										
7		PLIST	LITERALLY										
7		PHODRM	LITERALLY										
7		PNOFILIST	LITERALLY										
7		PNOLIST	LITERALLY										
7		PNOSEGFIX	LITERALLY										
7		PORG	LITERALLY										
7		PPAGESIZE	LITERALLY										
7		PPAGEWIDTH	LITERALLY										
7		PRB	LITERALLY										
127	0000H	240 PREFIX	BYTE ARRAY(240) EXTERNAL(75)	138	163	417							
127	0000H	1 PREFIXPTR	BYTE EXTERNAL(75)	135	139	154	416						
7		PREL9	LITERALLY										
7		PRELW	LITERALLY										
50	0000H	PRINTCRLF	PROCEDURE EXTERNAL(13) STACK=0000H										
127	0000H	1 PRINTDEVICE	BYTE EXTERNAL(68)										
56	0000H	PRINTINTI	PROCEDURE EXTERNAL(16) STACK=0000H										
52	0000H	PRINTNEWPAGE	PROCEDURE EXTERNAL(14) STACK=0000H										
127	0000H	1 PRINTON	BYTE EXTERNAL(62)										
47	0000H	PRINTSINGLYTE	PROCEDURE EXTERNAL(12) STACK=0000H										
54	0000H	PRINTSOURCELINE	PROCEDURE EXTERNAL(15) STACK=0000H										
127	0000H	1 PRINTSWITCHOFF	BYTE EXTERNAL(63)										
58	0000H	PRINTTERMINATE	PROCEDURE EXTERNAL(17) STACK=0000H										
2		PROC	LITERALLY	26	29	38	41	44	47	50			
				52	54	56	58	61	63	65	67	69	72
				75	79	81	83	86	89	92	95	93	101
				104	107	109	111	114	117	120	122	124	130
				135	142	157	159	225	228	230	234	243	265
				277	335	338	341	344	349	352	355	358	361
				364	366								
7		PRS	LITERALLY										
7		PRW	LITERALLY										
7		PSEGFIX	LITERALLY										
4		PSEUDO	LITERALLY										
1	0000H	PSEUDOM	PROCEDURE STACK=0000H										
9		PSEUDOOPEPRK	LITERALLY	154	223	412							
7		PSIMFORM	LITERALLY										
7		PSSEG	LITERALLY										
17	0000H	2 PT	WORD PARAMETER	18									
114	0000H	2 PT	WORD PARAMETER	115									
111	0000H	2 PT	WORD PARAMETER	112									
7		PTITLE	LITERALLY										
101	0000H	2 PTR	WORD PARAMETER	102									
6		RAH	LITERALLY										
6		RAL	LITERALLY										
6		RAX	LITERALLY										
6		RAH	LITERALLY										
6		RBL	LITERALLY										
6		RBP	LITERALLY										
6		RBX	LITERALLY										
6		RCH	LITERALLY										

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93350

SER. # _____

6		RCL	LITERALLY	
6		RCS	LITERALLY	356
6		RCX	LITERALLY	
6		RDH	LITERALLY	
6		RD1	LITERALLY	
6		RDL	LITERALLY	
6		RDS	LITERALLY	355
6		RDX	LITERALLY	
2		REENT	LITERALLY	
4		REG	LITERALLY	
6		RES	LITERALLY	359
98	0000H	2 RESULT	WORD PARAMETER	99
89	0000H	2 RESULT	WORD PARAMETER	90
83	0000H	2 RESULT	WORD PARAMETER	84
104	0000H	2 RETURNPTR	WORD PARAMETER	105
8		RIGHTBRACKET	LITERALLY	
6		RSI	LITERALLY	
6		RSP	LITERALLY	
344	03A8H	20 RSROUT	PROCEDURE PUBLIC STACK=0014H	
6		RSS	LITERALLY	356
35	0000H	2 S	WORD PARAMETER	36
127	0000H	12 SAVESOURCE	BYTE ARRAY(12) EXTERNAL(67)	
65	0000H	SCAN	PROCEDURE EXTERNAL(20) STACK=0000H	205 252
			308 406	
63	0000H	SCANINIT	PROCEDURE EXTERNAL(19) STACK=0000H	
229	0005H	1 SEGHIGH	BYTE EXTERNAL(97) AT	287
229	0004H	1 SEGLOW	BYTE EXTERNAL(97) AT	286
3		SEGBIT	LITERALLY	282
3		SEGHCOUNT	LITERALLY	
365	0004H	2 SEGMENT	WORD MEMBER(OPER)	
229	0004H	2 SEGMENT	WORD MEMBER(OPER)	229
158	0004H	2 SEGMENT	WORD MEMBER(OPER)	
143	0004H	2 SEGMENT	WORD MEMBER(OPER)	
127	0004H	2 SEGMENT	WORD MEMBER(OPERANDS)	
127	0004H	2 SEGMENT	WORD MEMBER(CURRENTSYMBOL)	
157	0076H	274 SEGMENTROUT	PROCEDURE STACK=0000H	350 353 356 359
157	0000H	1 SEGR	BYTE PARAMETER	158 161 162 192 199 214
158	0000H	1 SEGSPECIFIED	BYTE BASED(P2)	200
3		SEGTYPEBIT	LITERALLY	
3		SEGTYPECOUNT	LITERALLY	
365	0003H	1 SFLAG	BYTE MEMBER(OPER)	
229	0003H	1 SFLAG	BYTE MEMBER(OPER)	282
158	0003H	1 SFLAG	BYTE MEMBER(OPER)	
143	0003H	1 SFLAG	BYTE MEMBER(OPER)	
127	0003H	1 SFLAG	BYTE MEMBER(OPERANDS)	
127	0003H	1 SFLAG	BYTE MEMBER(CURRENTSYMBOL)	
127	0000H	1 SIMFORM	BYTE EXTERNAL(72)	
67	0000H	SKIPRESTOFLINE	PROCEDURE EXTERNAL(21) STACK=0000H	133 155
			190 222 314 318 383 388 401 413	
72	0000H	SKIPUNTIL	PROCEDURE BYTE EXTERNAL(23) STACK=0000H	
95	0000H	2 SOURCE	WORD PARAMETER	96
127	0000H	80 SOURCEBUF	BYTE ARRAY(80) EXTERNAL(73)	
127	0000H	12 SOURCENAME	BYTE ARRAY(12) EXTERNAL(66)	
127	0000H	1 SOURCEPTR	BYTE EXTERNAL(74)	
2		SPACE	LITERALLY	
4		SPEC	LITERALLY	
69	0000H	SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(22) STACK=0000H	197

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

305
127 0000H 2 SSCTP. . . . . WORD EXTERNAL(52) 189 356
10 SSDATA . . . . . LITERALLY 231
355 03EAH 23 SSGROUT . . . . . PROCEDURE PUBLIC STACK=0010H
127 0000H 1 SSPEC. . . . . BYTE EXTERNAL(69) 179 356
10 SSVALUE . . . . . LITERALLY 160
10 STARTTYPE . . . . . LITERALLY
98 0000H 2 STRADR . . . . . WORD PARAMETER 99
89 0000H 2 STRADR . . . . . WORD PARAMETER 99
86 0000H 2 STRADR . . . . . WORD PARAMETER 87
83 0000H 2 STRADR . . . . . WORD PARAMETER 84
4 STRTNG . . . . . LITERALLY 244
2 STRUC. . . . . LITERALLY 127 143 158 229 365
365 0002H 1 STYPE. . . . . BYTE MEMBER(COPER)
229 0002H 1 STYPE. . . . . BYTE MEMBER(COPER) 237 251 230
158 0002H 1 STYPE. . . . . BYTE MEMBER(COPER) 211
143 0002H 1 STYPE. . . . . BYTE MEMBER(COPER) 146
127 0002H 1 STYPE. . . . . BYTE MEMBER(COPERANDS)
127 0002H 1 STYPE. . . . . BYTE MEMBER(CURRENTSYMBOL) 373 386 393
404 410
95 0000H 2 SYMBADR. . . . . WORD PARAMETER 96
92 0000H 2 SYMBADR. . . . . WORD PARAMETER 93
81 0000H SYMBINIT . . . . . PROCEDURE EXTERNAL(28) STACK=0000H
4 SYMBOL . . . . . LITERALLY
127 0000H 1 SYMBOLDEVICE . . . . . BYTE EXTERNAL(69)
3 SYMBOLHEAD . . . . . LITERALLY
3 SYMBOLSTRUC. . . . . LITERALLY 127 143 158 229 365
127 0000H 2 SYMBTABADR . . . . . WORD EXTERNAL(95) 374 382 387 398
79 0000H SYMBTERMINATE. . . . . PROCEDURE EXTERNAL(27) STACK=0000H
11 0000H 2 T. . . . . WORD PARAMETER 12
2 TAB. . . . . LITERALLY
130 0000H 23 TESTEMPTYLINE. . . . . PROCEDURE STACK=0006H 150 206 218 420
14 0000H 2 TEXTADR. . . . . WORD PARAMETER 15
2 THENDO . . . . . LITERALLY 136 144 146 188 197 209 211
235 244 255 266 278 280 282 306 312 371
380 393 396 404 410
127 0000H 30 TITLE. . . . . BYTE ARRAY(30) EXTERNAL(70)
72 0000H 1 TOK. . . . . BYTE PARAMETER 73
69 0000H 1 TOK. . . . . BYTE PARAMETER 70
127 0000H 4 TOKEN. . . . . STRUCTURE EXTERNAL(79) 244
2 TRUE . . . . . LITERALLY 215 239 309 325 376
344 0004H 1 TYP. . . . . BYTE PARAMETER AUTOMATIC 345 347
142 0004H 1 TYP. . . . . BYTE PARAMETER AUTOMATIC 143 149
124 0000H 1 TYP. . . . . BYTE PARAMETER 125
127 0000H 1 TYPE . . . . . BYTE MEMBER(TOKEN) 244
3 TYPEBIT. . . . . LITERALLY
3 TYPECOUNT. . . . . LITERALLY
4 UDFSYMB . . . . . LITERALLY 386
32 0000H UPPER. . . . . PROCEDURE BYTE EXTERNAL(7) STACK=0000H
58 0000H 1 USEFACT. . . . . BYTE PARAMETER 59
127 0002H 2 VALUE. . . . . WORD MEMBER(TOKEN)
4 VARIABLE . . . . . LITERALLY 238
5 WRD. . . . . LITERALLY

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

CODE AREA SIZE = 052EH 1326D
CONSTANT AREA SIZE = 0008H 8D
VARIABLE AREA SIZE = 000EH 14D
MAXIMUM STACK SIZE = 001EH 30D
963 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE PSEUDO:

OBJECT MODULE PLACED IN PSEUD2.OBJ

COMPILER INVOKED BY: :F0: PSEUD2.PLN DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

```

1      $title ("PSEUDO INSTRUCTION MODULE-2")
      pseudom:
      do:

      /*

      modified 3/28/81 R. Silberstein
      modified 4/1/81 R. Silberstein
      modified 4/9/81 R. Silberstein
      modified 4/15/81 R. Silberstein
      modified 7/24/81 R. Silberstein
      modified 9/2/81 R. Silberstein

      */

      /*
      This is the module to perform the decoding of
      all legal pseudo instructions of the assembler.
      There is one subroutine for each corresponding
      pseudoinstruction.

      */

      $include (:f1:macro.lit)
= $nolist
      $include (:f1:struc.lit)
= $nolist
      $include (:f1:equals.lit)
= $nolist
      $include (:f1:ermod.lit)
= $nolist
      $include (:f1:files.ext)
= $nolist
      $include (:f1:subr1.ext)
= $nolist
      $include (:f1:subr2.ext)
= $nolist
      $include (:f1:scan.ext)
= $nolist
      $include (:f1:print.ext)
= $nolist
      $include (:f1:expr.ext)
= $nolist
      $include (:f1:ermod.ext)
= $nolist
      $include (:f1:pseud2.x86)
= $nolist

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$eject
/***** COMMON SUBROUTINES *****/

/* routine to test if rest of line is either a comment or empty -
   if not, print error message - skip rest of line */

139 1 test$emptyline: proc;
140 2     if not emptyline then call errmsg(end$of$line$err);
142 2     call skip$rest$of$line;
143 2 end test$emptyline;

/* perform handling for PAGEWIDTH- and PAGE$IZE-routine */

144 1 sizewidth: proc(p);
145 2     dcl oper operandstruc at (.operands(0)),p address,dest based p byte;
146 2     if pass=0 then$do
148 3         call skip$rest$of$line; /* do nothing in pass 0 */
149 3     else$do
151 3         if expression(.oper) then$do
153 4             if oper.stype = number then$do
155 5                 dest=oper.offset;
156 5                 call test$emptyline;
157 5                 return;
158 5             end$if;
159 5         end$if;
160 3         call errmsg(pseudooperr);
161 3         call skip$rest$of$line;
162 3     end$if;
163 2 end sizewidth;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

*reject

/***** PSEUDO SUBROUTINES *****/

```

164 1  !Froot: proc public;
165 2     DECLARE IFNESTMAX LIT "5"; /* MAX LEVEL OF IF NESTING */
166 2     dcl oper operandstruc at (.operands(0)),bool byte;

167 2  !Ferr: proc;
168 3     call errmsg(ifparerr);
169 3     call skip$rest$of$line;
170 3  end !Ferr;

171 2  skip$until$ENDIF: proc;
172 3     dcl pseudotype byte at (.token.value);
173 3     DECLARE LOCIFLEVEL BYTE;
174 3     DECLARE TEMP BYTE;
175 3     LOCIFLEVEL = IFLEVEL + 1;
176 3     IF PRINTON AND NOT IFLIST THEN$DO
177 4         CALL PRINTSOURCELINE;
178 4         PRINTON = FALSE;
179 4         TEMP = TRUE;
180 4     ELSE$DO
181 4         TEMP = FALSE;
182 4     END$IF;
183 3     do while not eofset; /* (forever) */
184 4         call scan;
185 4         IF TOKEN.TYPE = PSEUDO THEN$DO
186 5             IF PSEUDOTYPE = PENDIF THEN$DO
187 6                 LOCIFLEVEL = LOCIFLEVEL - 1;
188 6                 IF LOCIFLEVEL = 0 THEN$DO
189 7                     CALL SCAN;
190 7                     CALL TESTEMPTYLINE;
191 7                     IF TEMP THEN PRINTON = TRUE;
192 7                     RETURN;
193 7                 END$IF;
194 5             ELSE$DO
195 6                 IF PSEUDOTYPE = PIF THEN$DO
196 7                     LOCIFLEVEL = LOCIFLEVEL + 1;
197 7                 END$IF;
198 5             END$IF;
199 4         END$IF;
200 4         call skip$rest$of$line;
201 4     end$while;
202 3  end skip$until$ENDIF;

211 2  IF IFLEVEL = IFNESTMAX THEN$DO
212 3     call errmsg(nestediferr);
213 3     call skip$rest$of$line;
214 3  else$do
215 3     if not noforwardexpr(.oper) then$do
216 4         call !Ferr;
217 4     else$do
218 5         if oper.stype <> number then$do
219 6             call !Ferr;
220 6         else$do
221 7             bool=oper.offset;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

228 5      if bool <> 0 then$do
230 6          IFLEVEL = IFLEVEL + 1;
231 6          call test$emptyline;
232 6      else$do
234 6          call skip$rest$of$line;
235 6          call skip$until$ENDIF;
236 6      endif;
237 5      end$if;
238 4      end$if;
239 3      end$if;
240 2      end if$out;

241 1      ENDIF$out: proc public;
242 2          IF IFLEVEL > 0 THEN$DO
244 3              IFLEVEL = IFLEVEL - 1;
245 3              call test$emptyline;
246 3          else$do
248 3              call errmsg(missiferr);
249 3              call skip$rest$of$line;
250 3          end$if;
251 2      end ENDIF$out;

252 1      INCLUDE$out: proc public;
253 2          dcl (disk,i,errflag) byte,filename(11) byte,filtype(3) byte at (.filename (8));

254 2          syntaxerr: proc;
255 3              call errmsg(filesynterr);
256 3              errflag=true;
257 3          end syntaxerr;

258 2          accum$not$alpha: proc byte;
259 3              i=0ffh;
260 3              do while (i:=i+1) < acclen;
261 4                  if not alphanumeric(accum(i)) then return true;
264 3              return false;
265 3          end accum$not$alpha;

266 2          if include$on then$do
268 3              call errmsg(nestedincludeerr);
269 3              call skip$rest$of$line;
270 3              return;
271 3          end$if;

272 2          errflag=false;
273 2          disk=include$default; /* default disk is current one */
274 2          CALL FILL (SPACE, SIZE (FILENAME), .FILENAME);

275 2          if (acclen=1) and (nextch=':') and (letter(accum(0))) then$do

              /* disk name found */
277 3              disk=accum(0)-"A";
278 3              call scan; /* skip : */
279 3              call scan; /* get filename */
280 3          end$if;

              /* test syntax of filename */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

281 2      if (acclen > 8) or accum$not$alpha then$do
283 3          call syntaxerr; /* illegal filename */
284 3      else$do
286 3          call copy(acclen,.accum(0),.filename); /* pick up filename */
287 3          call scan; /* skip filename */

/* test if filetype - if so, pick it up */
288 3      if specialtoken(".") then$do
290 4          call scan; /* skip . */
291 4          if (acclen > 3) or accum$not$alpha then$do
293 5              call syntaxerr;
294 5          else$do
296 5              call copy(acclen,.accum(0),.filetype(0));
297 5              call scan;
298 5          end$if;
299 4      ELSE$DO
301 4          CALL COPY (3, .("A86"), .FILETYPE); /* DEFAULT FILE TYPE */
302 4          end$if;
303 3      end$if;

304 2      if errflag then$do
306 3          call skip$rest$of$line;
307 3      else$do
/* try to open include file */
309 3          call ifile$setup(disk,.filename,.filetype);
310 3          CALL OPEN$INCLUDE;
311 3          call test$emptyline;
312 3          include$on=true;
313 3          end$if;
314 2      end INCLUDE$out;

315 1      END$out: proc public;
316 2          call test$emptyline;
317 2          eofset=true;
318 2      end END$out;

319 1      PAGE$SIZE$out: proc public;
320 2          call sizewidth(.pagesize);
321 2      end PAGE$SIZE$out;

322 1      PAGE$WIDTH$out: proc public;
323 2          call sizewidth(.maxcol);
324 2      end PAGE$WIDTH$out;

325 1      TITLE$out: proc public;
326 2          do case pass;
327 3              do; /* pass 0 */
328 4                  if token.type=string then$do
330 5                      call fill(0,length(title),.title(0));
331 5                      if acclen > length(title) then acclen=length(title);
333 5                      call copy(acclen,.accum(0),.title(0));
334 5                  end$if;
335 4                  call skip$rest$of$line;
336 4                  end;
337 3              do; /* do nothing in pass 1 */
338 4                  call skip$rest$of$line;
339 4                  end;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
340 3      do;          /* pass 2 */
341 4      if token.type=string then$do
343 5          call scan;
344 5          call test$emptyline;
345 5      else$do
347 5          call errmsg(pseudooperr);
348 5          call skip$rest$of$line;
349 5      end$if;
350 4      end;
351 3      end$case;
352 2      end TITL$trout;

353 1      EJECT$trout: proc public;
354 2          if print$on then call print$newpage;
356 2          call test$emptyline;
357 2      end EJECT$trout;

358 1      SIMFORM$trout: proc public;
359 2          simform=true;
360 2          call test$emptyline;
361 2      end SIMFORM$trout;

362 1      LIST$trout: proc public;
363 2          call test$emptyline;
364 2          if print$switchoff then$do
366 3              print$switchoff=false;
367 3              print$on=true;
368 3              sourceptr=0;
369 3          end$if;
370 2      end LIST$trout;

371 1      NOLIST$trout: proc public;
372 2          if print$on then$do
374 3              call test$emptyline;
375 3              call print$source$line;
376 3              print$switchoff=true;
377 3              print$on=false;
378 3          else$do
380 3              call test$emptyline;
381 3          end$if;
382 2      end NOLIST$trout;

383 1      IFLISTROUT: PROC PUBLIC;
384 2          CALL TESTEMPTYLINE;
385 2          IFLIST = TRUE;
386 2      END IFLISTROUT;

387 1      NOIFLISTROUT: PROC PUBLIC;
388 2          CALL TESTEMPTYLINE;
389 2          IFLIST = FALSE;
390 2      END NOIFLISTROUT;

391 1      end$module pseudom;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
138	0000H	1	ACCLN BYTE EXTERNAL(52) 260 275 281 286 291 296 331 332 333
138	0000H	80	ACCU. BYTE ARRAY(80) EXTERNAL(63) 261 275 277 286 296 333
258	0240H	47	ACCU NOTALPHA. . . PROCEDURE BYTE STACK=0006H 291 291
2			ADDR LITERALLY 46 51 54 57 60 66 84 87 90 93 138 145 166
74	0000H		ALPHANUMERIC . . . PROCEDURE BYTE EXTERNAL(26) STACK=0000H 261
77	0000H		ASCII CHAR. PROCEDURE BYTE EXTERNAL(27) STACK=0000H
166	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
138	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
145	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
3			BASEFGBIT LITERALLY
3			BASEFGBIT LITERALLY
166	0008H	1	BOOL BYTE 227 228
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
5			BYT. LITERALLY
16	0000H	1	CH BYTE PARAMETER 17
65	0000H	1	CH BYTE PARAMETER 66
13	0000H	1	CH BYTE PARAMETER 14
10	0000H	1	CH BYTE PARAMETER 11
71	0000H	1	CH BYTE PARAMETER 72
68	0000H	1	CH BYTE PARAMETER 69
109	0000H	1	CH BYTE PARAMETER 110
74	0000H	1	CH BYTE PARAMETER 75
77	0000H	1	CH BYTE PARAMETER 78
80	0000H	1	CH BYTE PARAMETER 81
39	0000H		CLOSE HEX PROCEDURE EXTERNAL(13) STACK=0000H
37	0000H		CLOSE INCLUDE PROCEDURE EXTERNAL(12) STACK=0000H
41	0000H		CLOSE PRINT PROCEDURE EXTERNAL(14) STACK=0000H
33	0000H		CLOSE SOURCE PROCEDURE EXTERNAL(10) STACK=0000H
43	0000H		CLOSE SYMBOL. PROCEDURE EXTERNAL(15) STACK=0000H
4			CODE LITERALLY
9			CODEMACROERR. LITERALLY
3			CODEMACROHEAD. LITERALLY
56	0000H		COPY PROCEDURE EXTERNAL(20) STACK=0000H 286 296 301 333
2			CR LITERALLY
83	0000H	2	D. WORD PARAMETER 84
86	0000H	2	D. WORD PARAMETER 87
56	0000H	2	D. WORD PARAMETER 57
89	0000H	2	D. WORD PARAMETER 90
92	0000H	2	D. WORD PARAMETER 93
2			DCL. LITERALLY
92	0000H		DECOU. PROCEDURE EXTERNAL(32) STACK=0000H
4			DELETED SYMB. LITERALLY
138	0001H	1	DESCR. BYTE MEMBER(TOKEN)
145	0000H	1	DEST BYTE BASED(P) 155

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

45	0000H	1	DEV.	BYTE PARAMETER	46	
68	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(24) STACK=0000H		
253	0003H	1	DISK	BYTE	273 277 309	
4			DOUBLEDEFINED. . .	LITERALLY		
9			DOUBLEDEFVAR. . .	LITERALLY		
9			DOUBLEDEFLAB. . .	LITERALLY		
9			DOUBLEDEFSYM. . .	LITERALLY		
9			DOUBLEDEFVAR. . .	LITERALLY		
5			DWRD	LITERALLY		
123	0000H		EJECT.	PROCEDURE EXTERNAL(45) STACK=0000H		
353	02F5H	18	EJECTROUT.	PROCEDURE PUBLIC STACK=000AH		
2			ELSEDD.	LITERALLY	199	
107	0000H		EMPTYLINE.	PROCEDURE BYTE EXTERNAL(38) STACK=0000H		140
2			ENDCASE.	LITERALLY		
2			ENDDD.	LITERALLY		
2			ENDFOREVER.	LITERALLY		
2			ENDIF.	LITERALLY	158 205	
241	0119H	30	ENDIFROUT.	PROCEDURE PUBLIC STACK=000AH		
2			ENDMODULE.	LITERALLY		
2			ENDOFFILE.	LITERALLY		
9			ENDOFFLINEERR. . .	LITERALLY	141	
2			ENDPROC.	LITERALLY		
315	026FH	13	ENDROUT.	PROCEDURE PUBLIC STACK=000AH		
2			ENDWHILE.	LITERALLY	262	
138	0000H	1	EOFSET.	BYTE EXTERNAL(64)	185 317	
83	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(29) STACK=0000H		
253	0005H	1	ERRFLAG.	BYTE	256 272 304	
135	0000H		ERRMSG.	PROCEDURE EXTERNAL(50) STACK=0000H		141 160
					168 213 248 255 268 347	
135	0000H	1	ERRND.	BYTE PARAMETER	136	
4			ERROR.	LITERALLY		
129	0000H		EXPRESSION.	PROCEDURE BYTE EXTERNAL(48) STACK=0000H		151
2			FALSE.	LITERALLY	179 183 264 272 366	377 389
62	0000H	2	FCBADR.	WORD PARAMETER	63	
62	0000H		FILEABORT.	PROCEDURE EXTERNAL(22) STACK=0000H		
3			FILEISTRUCTURE. . .	LITERALLY		
3			FILEDSTRUCTURE. . .	LITERALLY		
48	0000H		FILESETUP.	PROCEDURE BYTE EXTERNAL(17) STACK=0000H		
9			FILESYNTERK.	LITERALLY	255	
65	0000H		FILL.	PROCEDURE EXTERNAL(23) STACK=0000H		274 330
45	0000H	2	FILNAM.	WORD PARAMETER	46	
253	0006H	11	FILNAME.	BYTE ARRAY(11)	253 274 286 309	
45	0000H	2	FILTYP.	WORD PARAMETER	46	
253	000EH	3	FILTYP.	BYTE ARRAY(3) AT	296 301 309	
9			FIRSTITEM.	LITERALLY		
2			FOREVER.	LITERALLY		
2			FORMFEED.	LITERALLY		
86	0000H		HEX1OUT.	PROCEDURE EXTERNAL(30) STACK=0000H		
89	0000H		HEX2OUT.	PROCEDURE EXTERNAL(31) STACK=0000H		
253	0004H	1	I.	BYTE	259 260 261	
4			IDENT.	LITERALLY		
167	0099H	14	IFERR.	PROCEDURE STACK=0006H	219 224	
45	0000H		IFLASETUP.	PROCEDURE EXTERNAL(16) STACK=0000H		309
138	0000H	1	IFLEVEL.	BYTE EXTERNAL(66)	175 211 230 242 244	
138	0000H	1	IFLIST.	BYTE EXTERNAL(54)	176 385 389	
383	0353H	13	IFLISTROUT.	PROCEDURE PUBLIC STACK=000AH		
165			IFWESTMAX.	LITERALLY	211	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

9			IFPAKERR	LITERALLY	168	
164	0040H	76	IFROUT	PROCEDURE PUBLIC STACK=000CH		
9			ILLEGALMACRO	LITERALLY		
9			ILLEGALPSEUDO	LITERALLY		
9			ILLEXPRELEM	LITERALLY		
9			ILLTOPER	LITERALLY		
138	0000H	1	INCLUDEDEFAULT	BYTE EXTERNAL(63)	273	
138	0000H	1	INCLUDEON	BYTE EXTERNAL(65)	265 312	
252	0137H	249	INCLUDEROUT	PROCEDURE PUBLIC STACK=000CH		
3			INDEXREGBIT	LITERALLY		
3			INDEXREGCOUNT	LITERALLY		
21	0000H		ININCLUDEBYTE	PROCEDURE BYTE EXTERNAL(4) STACK=0000H		
2			INIT	LITERALLY		
19	0000H		INSOURCEBYTE	PROCEDURE BYTE EXTERNAL(3) STACK=0000H		
9			INSTKERR	LITERALLY		
3			IREGBIT	LITERALLY		
3			IREGCOUNT	LITERALLY		
4			LAB	LITERALLY		
9			LABOUTOFRANGE	LITERALLY		
8			LEFTBRACKET	LITERALLY		
145	0000H	2	LENGTH	WORD MEMBER(OPER)		
138	0000H	2	LENGTH	WORD MEMBER(OPERANDS)		
166	0000H	2	LENGTH	WORD MEMBER(OPER)		
			LENGTH	BUILTIN	330 331 332	
71	0000H		LETTER	PROCEDURE BYTE EXTERNAL(25) STACK=0000H	275	
2			LF	LITERALLY		
362	0314H	30	LISTROUT	PROCEDURE PUBLIC STACK=0000H		
2			LIT	LITERALLY	2 3 4 5 6 7 8	
				9 165		
173	0001H	1	LOCIFLLVEL	BYTE	175 191 192 204	
138	0000H	1	MAXCOL	BYTE EXTERNAL(55)	323	
9			MISSIFERR	LITERALLY	248	
9			MISSINGPSEUDO	LITERALLY		
9			MISSANSTR	LITERALLY		
9			MISSSEGMINF	LITERALLY		
9			MISSYPEINFO	LITERALLY		
92	0000H	2	N	WORD PARAMETER	93	
89	0000H	2	N	WORD PARAMETER	90	
86	0000H	1	N	BYTE PARAMETER	87	
83	0000H	1	N	BYTE PARAMETER	84	
65	0000H	1	N	BYTE PARAMETER	66	
56	0000H	1	N	BYTE PARAMETER	57	
53	0000H	2	N	WORD PARAMETER	54	
4			NEGLECTED	LITERALLY		
9			NEGLECTERR	LITERALLY		
9			NESTEDIFERR	LITERALLY	213	
9			NESTEDINCLUDERR	LITERALLY	268	
138	0000H	1	NEXTCH	BYTE EXTERNAL(61)	275	
5			NIL	LITERALLY		
132	0000H		NOFORWARDEXPR	PROCEDURE BYTE EXTERNAL(49) STACK=0000H	217	
127	0000H		NOFORWARDOPER	PROCEDURE BYTE EXTERNAL(47) STACK=0000H		
387	0360H	13	NOIFLISTROUT	PROCEDURE PUBLIC STACK=0000H		
371	0332H	33	NOLISTROUT	PROCEDURE PUBLIC STACK=0000H		
3			NOOVERCOUNT	LITERALLY		
3			NOOVERRIDEIT	LITERALLY		
4			NUMBER	LITERALLY	153 222	
8			ORAND	LITERALLY		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

8		OEQ.	LITERALLY	
145	0006H	2 OFFSET	WORD MEMBER(COPER)	155
138	0006H	2 OFFSET	WORD MEMBER(COPERANDS)	
166	0006H	2 OFFSET	WORD MEMBER(COPER)	227
8		UGE.	LITERALLY	
8		UGI.	LITERALLY	
8		OLAST.	LITERALLY	
8		OLE.	LITERALLY	
8		OLENGTH.	LITERALLY	
8		OLT.	LITERALLY	
8		OMOD	LITERALLY	
8		ONE.	LITERALLY	
8		ONOT	LITERALLY	
8		OOFFSET.	LITERALLY	
8		ODR.	LITERALLY	
27	0000H	OPENEX.	PROCEDURE EXTERNAL(7) STACK=0000H	
25	0000H	OPENINCLUDE.	PROCEDURE EXTERNAL(6) STACK=0000H	310
29	0000H	OPENPRINT.	PROCEDURE EXTERNAL(8) STACK=0000H	
23	0000H	BPENSOURCE	PROCEDURE EXTERNAL(5) STACK=0000H	
31	0000H	OPENSMBOL	PROCEDURE EXTERNAL(9) STACK=0000H	
145	0000H	9 OPER	STRUCTURE EXTERNAL(67) AT	151 153 155
166	0000H	9 OPER	STRUCTURE EXTERNAL(67) AT	217 222 227
125	0000H	OPERAND.	PROCEDURE BYTE EXTERNAL(46) STACK=0000H	
138	0000H	36 OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(67)	145 166
3		OPERANDSTRUC	LITERALLY	138 145 166
4		OPERATOR	LITERALLY	
9		OPMISMATCH	LITERALLY	
8		OPTR	LITERALLY	
8		OSEG	LITERALLY	
8		OSHL	LITERALLY	
8		OSHORT	LITERALLY	
8		OSHR	LITERALLY	
8		OTYPE.	LITERALLY	
10	0000H	OUTHEXBYTE	PROCEDURE EXTERNAL(0) STACK=0000H	
13	0000H	OUTPRINTBYTE	PROCEDURE EXTERNAL(1) STACK=0000H	
16	0000H	OUTSYMBOLBYTE.	PROCEDURE EXTERNAL(2) STACK=0000H	
59	0000H	OUTTEXT.	PROCEDURE EXTERNAL(21) STACK=0000H	
8		OXOR	LITERALLY	
144	0004H	2 P.	WORD PARAMETER AUTOMATIC	145 155
138	0000H	1 PAGESTZE	BYTE EXTERNAL(57)	320
319	027CH	12 PAGESIZEROUT	PROCEDURE PUBLIC STACK=0010H	
322	0286H	12 PAGEWIDTHROUT.	PROCEDURE PUBLIC STACK=0010H	
138	0000H	1 PASS	BYTE EXTERNAL(51)	146 326
7		PCUDEMACRO	LITERALLY	
7		PCSEG.	LITERALLY	
7		PDB.	LITERALLY	
7		PDBIT.	LITERALLY	
7		PDD.	LITERALLY	
7		PDSEG.	LITERALLY	
7		PDW.	LITERALLY	
7		PEJECT	LITERALLY	
7		PEND	LITERALLY	
7		PENDIF	LITERALLY	189
7		PENDM.	LITERALLY	
7		PEQU	LITERALLY	
7		PESEG.	LITERALLY	
7		PIF.	LITERALLY	202

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

7		PIFLIST.	LITERALLY						
7		PINCLUDE.	LITERALLY						
7		PLIST.	LITERALLY						
7		PMDRM.	LITERALLY						
7		PNOIFLIST.	LITERALLY						
7		PNOLIST.	LITERALLY						
7		PNOSEGFIX.	LITERALLY						
7		PORG.	LITERALLY						
7		PPAGESIZE.	LITERALLY						
7		PPAGEWIDTH.	LITERALLY						
7		PRB.	LITERALLY						
7		PRELB.	LITERALLY						
7		PRELW.	LITERALLY						
112	0000H	PRINTCRLF.	PROCEDURE EXTERNAL(40) STACK=0000H						
118	0000H	PRINTINIT.	PROCEDURE EXTERNAL(43) STACK=0000H						
114	0000H	PRINTNEWPAGE.	PROCEDURE EXTERNAL(41) STACK=0000H				355		
138	0000H	1 PRINTON.	BYTE EXTERNAL(52)	176	179	197	354	367	
			372 377						
109	0000H	PRINTSINGLEBYTE.	PROCEDURE EXTERNAL(39) STACK=0000H						
116	0000H	PRINTSOURCELINE.	PROCEDURE EXTERNAL(42) STACK=0000H				178	375	
138	0000H	1 PRINTSWITCHOFF.	BYTE EXTERNAL(53)	364	366	376			
120	0000H	PRINTITERMATE.	PROCEDURE EXTERNAL(44) STACK=0000H						
2		PROC.	LITERALLY	10	13	16	19	21	23
				27	29	31	33	35	37
				48	50	53	74	77	86
				99	101	104	107	109	112
				123	125	127	129	132	135
				171	241	252	254	258	315
				358	362	371	383	387	
7		PRS.	LITERALLY						
7		PRW.	LITERALLY						
7		PSEGFIX.	LITERALLY						
4		PSEUDO.	LITERALLY	187					
1	0000H	PSEUDOM.	PROCEDURE STACK=0000H						
9		PSEUDOOPERR.	LITERALLY	160	347				
172	0002H	1 PSEUDOTYPE.	BYTE EXTERNAL(60) AT	189	202				
7		PSIMFORM.	LITERALLY						
7		PSSEG.	LITERALLY						
132	0000H	2 PT.	WORD PARAMETER	133					
129	0000H	2 PT.	WORD PARAMETER	130					
65	0000H	2 PT.	WORD PARAMETER	66					
7		PTITLE.	LITERALLY						
6		RAH.	LITERALLY						
6		RAL.	LITERALLY						
6		RAX.	LITERALLY						
6		RBH.	LITERALLY						
6		RBL.	LITERALLY						
6		RBP.	LITERALLY						
6		RBX.	LITERALLY						
6		RCH.	LITERALLY						
6		RCL.	LITERALLY						
6		RCS.	LITERALLY						
6		RCX.	LITERALLY						
6		RDI.	LITERALLY						
6		RDL.	LITERALLY						
6		RDS.	LITERALLY						

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

6		RDX.	LITERALLY	
2		RENT.	LITERALLY	
4		REG.	LITERALLY	
6		RES.	LITERALLY	
35	0000H	REWINDSOURCE	PROCEDURE EXTERNAL(11) STACK=0000H	
8		RIGHTBRACKET	LITERALLY	
6		RSI.	LITERALLY	
6		RSP.	LITERALLY	
6		RSS.	LITERALLY	
83	0000H	2 S.	WORD PARAMETER 84	
56	0000H	2 S.	WORD PARAMETER 57	
97	0000H	SCAN	PROCEDURE EXTERNAL(34) STACK=0000H	186 194
			278 279 287 290 297 343	
95	0000H	SCANINIT	PROCEDURE EXTERNAL(35) STACK=0000H	
3		SEGMENT.	LITERALLY	
3		SEGMCOUNT.	LITERALLY	
166	0004H	2 SEGMENT.	WORD MEMBER(OPER)	
145	0004H	2 SEGMENT.	WORD MEMBER(OPER)	
138	0004H	2 SEGMENT.	WORD MEMBER(OPERANDS)	
3		SEGTYPEBIT	LITERALLY	
3		SEGTYPECOUNT	LITERALLY	
166	0003H	1 SFLAG.	BYTE MEMBER(OPER)	
145	0003H	1 SFLAG.	BYTE MEMBER(OPER)	
138	0003H	1 SFLAG.	BYTE MEMBER(OPERANDS)	
138	0000H	1 SIMFORM.	BYTE EXTERNAL(58) 359	
358	0307H	13 SIMFORMROUT. . . .	PROCEDURE PUBLIC STACK=000AH	
			SIZE 274	
144	0017H	54 SIZEWIDTH.	PROCEDURE STACK=000CH 320 323	
99	0000H	SKIPRESTOFLINE	PROCEDURE EXTERNAL(35) STACK=0000H	142 148
			161 169 208 214 234 249 269 306 335 338	
			348	
104	0000H	SKIPUNTIL.	PROCEDURE BYTE EXTERNAL(37) STACK=0000H	
171	00A7H	114 SKIPUNTILENDIF . . .	PROCEDURE STACK=000AH 235	
138	0000H	1 SOURCEPTR.	BYTE EXTERNAL(59) 368	
2		SPACE.	LITERALLY 274	
4		SPEC.	LITERALLY	
101	0000H	SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(36) STACK=0000H	288
4		STRING.	LITERALLY 328 341	
2		STRUC.	LITERALLY 138 145 166	
166	0002H	1 STYPE.	BYTE MEMBER(OPER) 222	
145	0002H	1 STYPE.	BYTE MEMBER(OPER) 153	
138	0002H	1 STYPE.	BYTE MEMBER(OPERANDS)	
4		SYMBOL.	LITERALLY	
3		SYMBOLHEAD.	LITERALLY	
3		SYMBOLSTRUC.	LITERALLY 138 145 166	
254	0230H	16 SYNTAXERR.	PROCEDURE STACK=0006H 283 293	
59	0000H	2 T.	WORD PARAMETER 60	
2		TAB.	LITERALLY	
174	0002H	1 TEMP.	BYTE 180 183 196	
139	0000H	23 TESTEMPTYLINE. . . .	PROCEDURE STACK=0006H 156 195 231 245	
			311 316 344 356 360 363 374 380 384 388	
62	0000H	2 TEXTADR.	WORD PARAMETER 63	
2		THENDO	LITERALLY 146 151 153 176 187 189 192	
			202 211 217 222 228 242 266 275 281 288	
			291 304 328 341 364 372	
138	0000H	30 TITLE.	BYTE ARRAY(30) EXTERNAL(56) 330 331 332	
			333	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

325 0294H 97 TITLEROUT. . . . . PROCEDURE PUBLIC STACK=0000H
104 0000H 1 TOK. . . . . . . . . . . BYTE PARAMETER 105
101 0000H 1 TOK. . . . . . . . . . . BYTE PARAMETER 102
138 0000H 4 TOKEN. . . . . . . . . . . STRUCTURE EXTERNAL(60) 172 137 328 341
2 TRUE . . . . . . . . . . . LITERALLY 120 157 256 262 312 317 359
367 376 385
138 0000H 1 TYPE . . . . . . . . . . . BYTE MEMBER(TOKEN) 187 328 341
3 TYPEBIT. . . . . . . . . . . LITERALLY
50 0000H TYPECALC . . . . . . . . . . . PROCEDURE BYTE EXTERNAL(18) STACK=0000H
3 TYPECOUNT. . . . . . . . . . . LITERALLY
4 UDEFSYMB . . . . . . . . . . . LITERALLY
9 UDEFSYMBOL . . . . . . . . . . . LITERALLY
80 0000H UPPER. . . . . . . . . . . PROCEDURE BYTE EXTERNAL(28) STACK=0000H
120 0000H 1 USEFACT. . . . . . . . . . . BYTE PARAMETER 121
50 0000H 2 VAL. . . . . . . . . . . WORD PARAMETER 51
138 0002H 2 VALUE. . . . . . . . . . . WORD MEMBER(TOKEN) 172
4 VARIABLE . . . . . . . . . . . LITERALLY
5 WRD. . . . . . . . . . . LITERALLY
53 0000H WRDTEST. . . . . . . . . . . PROCEDURE BYTE EXTERNAL(19) STACK=0000H

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 0360H      877D
CONSTANT AREA SIZE  = 0003H       3D
VARIABLE AREA SIZE  = 0011H       17D
MAXIMUM STACK SIZE  = 0010H       16D
862 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

CCP/M-86 2.0 V.3

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104



ISIS-II PL/M-86 V2.0 COMPILETON OF MODULE C4SUBR

OBJECT MODULE PLACED IN CMSUBR.OBJ

COMPILER INVOKED BY: :F0: CMSUBR.PL4 DEBUG OPTIMIZE(2) DATE(10/5/81) PAGE1DTP(100) XREF

1 \$title ("CODEMACRO SUBROUTINE MODULE")

cmsubr:

do:

/*

modified 4/7/81 R. Silberstein

modified 4/13/81 R. Silberstein

modified 5/5/81 R. Silberstein

modified 9/2/81 R. Silberstein

*/

/*

This is the module to

1) test if a set of operands matches a given instruction

and

2) produce output code for matched instruction

The module interfaces the CODFOUTPUT module to
physically send code bytes to the HEX output file.

*/

```

= $include (:f1:macro.lit)
= $nolist
= $include (:f1:equals.lit)
= $nolist
= $include (:f1:cmacd.lit)
= $nolist
= $include (:f1:outp.lit)
= $nolist
= $include (:f1:scan.ext)
= $nolist
= $include (:f1:subr1.ext)
= $nolist
= $INCLUDE (:F1:SUBR2.EXI)
= $nolist
= $include (:f1:outp.ext)
= $nolist
= $include (:f1:ermod.ext)
= $nolist
= $include (:f1:cmsubr.x86)
= $nolist

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
90 1  $eject
      dcl /* global variables */
      bytevar based macroptr byte, /* variables within codemacros */
      addrvar based macroptr addr,
      emitbyte(80) byte, /* buffer of output codebytes */
      emitindex byte, /* index of "emitbyte" */
      bitcomtab(2) byte data /* legal commands within "DBIT" */
      (nnumberbits,mformatbits);
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

\$object

/***** MICHELLAREOUS SUBROUTINES: *****/

```

91 1  clearcindex: proc public;
92 2      emitindex=0;
93 2  end clearcindex;

94 1  emit: proc public; /* emit codebytes for an instruction */
95 2      dcl i byte;
96 2      i=0ffh;
97 2      do while (i:=i+1) < emitindex;
98 3          call emitcodebyte(emitbyte(i),CSdata);
99 3      end$while;
100 2  end emit;

101 1  emitdummies: proc public; /* emit dummy (NO-OP-) bytes if error */
102 2      dcl (i,j) byte,nodum(4) byte data(2,5,6,8);
103 2      j=nooper;
104 2      if j>3 then j=3;
105 2      i=0ffh;
106 2      do while (i:=i+1) < nodum(j);
107 3          call emitcodebyte(90h,CSdata); /* 90H = NOP */
108 3      end$while;
109 3  end emitdummies;
110 2  end emitdummies;

111 1  emitsinglebyte: proc(ch); /* fill local emitbuffer with a new byte */
112 2      dcl ch byte;
113 2      if noerror then$do
114 3          emitbyte(emitindex)=ch;
115 3          emitindex=emitindex+1;
116 3      end$if;
117 3  end emitsinglebyte;
118 2  end emitsinglebyte;

119 1  emitsingleword: proc (var); /* fill 2 new bytes into emitbuffer */
120 2      dcl var addr, byt1 byte at(.var), byt2 byte at(.var+1);
121 2      call emitsinglebyte(byt1);
122 2      call emitsinglebyte(byt2);
123 2  end emitsingleword;

124 1  incrmacroptr: proc;
125 2      macroptr=macroptr+1;
126 2  end incrmacroptr;

127 1  getoperadr: proc address;
128 2      dcl pt address;
129 2      pt=.operands(bytevar);
130 2      call incrmacroptr;
131 2      return pt;
132 2  end getoperadr;

/* recognize codemacro command type */
133 1  commandtype: proc(comno,lg,pt) byte public;
134 2      dcl (comno,lg,i) byte,pt address,ch based pt(1) byte;
135 2      i=0ffh;
136 2      do while (i:=i+1) < lg;
137 3          if comno=ch(i) then$do call incrmacroptr; return i; end$if;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
143 2      endwhile;  
144 2      return lg;  
      end commandtype;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$eject

/***** CODENACRO COMMAND SUBROUTINES: *****/

```

145 1  mDbNROUT: proc public;
146 2      call emitsinglebyte(bytevar);
147 2      call incrmacroptr;
148 2  end mDbNROUT;

149 1  mDbFrout: proc public;
150 2      dcl pt address,opr based pt operandstruc;
151 2      pt=getoperadr;
152 2      call emitsinglebyte(opr.offset);
153 2  end mDbFrout;

154 1  mDwNROUT: proc public;
155 2      call emitseingleword(addrvar);
156 2      call incrmacroptr;
157 2      call incrmacroptr;
158 2  end mDwNROUT;

159 1  mDwFrout: proc public;
160 2      dcl pt address,opr based pt operandstruc;
161 2      pt=getoperadr;
162 2      call emitseingleword(opr.offset);
163 2  end mDwFrout;

164 1  mDdFrout: proc public;
165 2      dcl pt address,opr based pt operandstruc;
166 2      pt=getoperadr;
167 2      if (opr.sflag and segmbit) = 0 then call errmsg(missesegminfo);
168 2      call emitseingleword(opr.offset);
169 2      call emitseingleword(opr.segment);
170 2      call emitseingleword(opr.segment);
171 2  end mDdFrout;

172 1  mRELBROUT: proc public;
173 2      dcl pt address,opr based pt operandstruc,displ addr;
174 2      pt=getoperadr;
175 2      displ=opr.offset-cip-2;
176 2      if (opr.segment <> csegvalue) or (typecalc(displ)=wrd) then$do
177 3          call errmsg(laboutofrange);
178 3      end$if;
179 3      call emitsinglebyte(displ);
180 2      IF ABSADDR (0) = SPACE THEN$DO
181 3          CALL HEX2OUT (OPR.OFFSET, .ABSADDR);
182 3      END$IF;
183 3      END$IF;
184 3      END$IF;
185 2  end mRELBROUT;

186 1  mRELWROUT: proc public;
187 2      dcl pt address,opr based pt operandstruc;
188 2      pt=getoperadr;
189 2      if opr.segment <> csegvalue then call errmsg(laboutofrange);
190 2      call emitseingleword(opr.offset-cip-3);
191 2      IF ABSADDR (0) = SPACE THEN$DO
192 3          CALL HEX2OUT (OPR.OFFSET, .ABSADDR);
193 3      END$IF;
194 3      END$IF;
195 2  end mRELWROUT;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

196 2      end mRELw rout;

197 1      mNOSECFIXrout: proc public;
198 2          dcl (segr,flag,segt) byte,pt address,opr based pt operandstruc;
199 2          segr=bytevar;
200 2          call incrmacroptr;
201 2          pt=getoperadr;
202 2          if (opr.baseindex and nooverridebit) = 0 then do
204 3              flag=opr.sflag;
205 3              segt=shr(flag,segtypecount) and 3;
206 3              noerror=(segt=segr);
207 3          end if;
208 2      end mNOSECFIXrout;

209 1      mSEGFIXrout: proc public;
210 2          dcl pt address,opr based pt operandstruc,(segr,override,sflag) byte;
211 2          DSovertest: proc byte;
212 3              segr=shr(opr.baseindex,basereqcount) and 1;
213 3              return (((sflag and bregbit) <> 0) and (segr=1)); /* 1 = BP */
214 3          end DSovertest;

215 2          pt=getoperadr;
216 2          sflag=opr.sflag;
217 2          if (opr.baseindex and nooverridebit) = 0 then do
219 3              segr=shr(sflag,segtypecount) and 3;
220 3              do case segr;
221 4                  do; override=true; segr=ESover; end; /* ES */
222 4                  do; override=true; segr=CSover; end; /* CS */
223 4                  do; override=not DSovertest; segr=SSover; end; /* SS */
224 4                  do; override=DSovertest; segr=DSover; end; /* DS */
225 4              end case;
226 4              if override then call emitsinglebyte(segr);
227 4          end if;
228 3      end mSEGFIXrout;

242 1      MODRM: proc (regfield,pt);
243 2          dcl pt address,opr based pt operandstruc,
                (regfield,modfield,rmfield,dispflag,stype,sflag,segr) byte,
                BASEIND BYTE,
                offset addr,
                displow byte at(.offset),disphigh byte at (.offset+1);

244 2          disptype: proc byte;
245 3              if segr=rce then return 2; /* disp always 2 for variable in CS */
247 3              if offset = 0 then return 0;
249 3              return typcalc(offset);
250 3          end disptype;

251 2          indextype: proc byte;
252 3              if (sflag and iregbit) <> 0 then do
254 4                  if (sflag and bregbit) <> 0 then return 0;
256 4                  return 1;
257 4              end if;
258 3              return 2;
259 3          end indextype;

260 2          offset=opr.offset; /* pick up operand attributes */

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

261 2      stype=opr.stype;
262 2      sflag=opr.sflag;
263 2      seg=shr(sflag,se;typecount) and 3;
264 2      BASEIND = OPR.BASEINDEX AND (BASEREGBIT OR INDEXREGBIT);

265 2      if stype=reg then$do
267 3          rmfield=offset;
268 3          modfield=11b;
269 3          dispflag=0;
270 3      else$do
272 3          if (sflag and (iregbit or bregbit)) = 0 then$do
274 4              rmfield=110b;
275 4              modfield=0;
276 4              dispflag=2;
277 4          else$do
279 4              dispflag=disptype; /* get no of DISP bytes */
280 4              modfield=dispflag;
281 4              do case indextype;

/* both base- and index-reg */
282 5          RMFIELD = BASEIND AND (INDEXREGBIT OR BASEREGBIT);

/* index reg only */
283 5          RMFIELD = 100B OR (BASEIND AND INDEXREGBIT);

284 5          do; /* base reg only */
285 6              IF (BASEIND AND BASEREGBIT) > 0 THEN$DO
287 7                  rmfield=110b;
/* mod=00 and r/m=110B is a special case */
288 7                  if dispflag=0 then$do
290 8                      dispflag,modfield=1;
291 8                  end$if;
                else$do
294 7                      rmfield=111b;
295 7                  end$if;
296 6                  end;
297 5              end$case;
298 4          end$if;
299 3      end$if;
300 2      regfield=shl(regfield,3) and 38h;
301 2      modfield=shl(modfield,6) and 0c0h;
302 2      call emitsinglebyte(regfield or modfield or rmfield);
303 2      if dispflag > 0 then$do
305 3          call emitsinglebyte(displow);
306 3          if dispflag=2 then call emitsinglebyte(disphigh);
                end$if;
309 2      end MODRM;

310 1      mMODRM1rout: proc public;
311 2          decl regfield byte;
312 2          regfield=bytevar;
313 2          call incrmacroptr;
314 2          call MODRM(regfield,getoperadr);
315 2          end mMODRM1rout;

316 1      mMODRM2rout: proc public;
317 2          decl regfield byte,pt address,opr based pt operandstruc;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

318 2      pt=getoperadr;
319 2      regfield=opr.offset;
320 2      call MODRM(regfield,getoperadr);
321 2      end mMODRM2rout;

322 1      mDBITrout: proc public;
323 2      decl (result,crbit) byte,bittab(8) byte data(1,2,4,8,16,32,64,128);

324 2      join: proc(numb,nobit,noshift);
325 3      decl (numb,nobit,noshift) byte;
326 3      if noshift > 0 then numb=shr(numb,noshift);
328 3      if nobit < 8 then numb=shl(numb,8-nobit);
330 3      do while (crbit <> 0ffh) and (nobit > 0);
331 4          if (numb and 80h) <> 0 then result=result or bittab(crbit);
333 4          crbit=crbit-1;
334 4          nobit=nobit-1;
335 4          numb=shl(numb,1);
336 4      end$while;
337 3      end join;

338 2      NUMBERBITSrout: proc;
339 3      decl nobit byte;
340 3      nobit=bytevar;
341 3      call incrmacroptr;
342 3      call join(bytevar,nobit,0);
343 3      call incrmacroptr;
344 3      end NUMBERBITSrout;

345 2      FORMBITSrout: proc;
346 3      decl (nobit,numb) byte,pt address,opr based pt operandstruc;
347 3      nobit=bytevar;
348 3      call incrmacroptr;
349 3      pt=getoperadr;
350 3      numb=opr.offset;
351 3      call join(numb,nobit,bytevar);
352 3      call incrmacroptr;
353 3      end FORMBITSrout;

354 2      result=0;
355 2      crbit=7;      /* current bit position */
356 2      do while bytevar <> mendbit; /* do until ENDBIT command */
357 3          do case commandtype(bytevar,length(bitcomtab),.bitcomtab);
358 4              call NUMBERBITSrout;
359 4              call FORMBITSrout;
360 4          do; end;
362 4          end$case;
363 3      end$while;
364 2      call incrmacroptr; /* skip ENDBIT command */
365 2      call emitsinglebyte(result);
366 2      end mDBITrout;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$object

/***** ROUTINES TO MATCH OPERANDS TO INSTRUCTION *****/

/* test user operand against codemacro parameter */

```
367 1 matchsingleop: proc(opno) byte;
368 2   dcl (match,specletter,modletter,range,rangetype) byte,
      (rangev1,rangev2,opno) byte,
      pt address, oper based pt operandstruct;
```

```
369 2   rangetest: proc byte; /* perform rangetest */
370 3     dcl opvalue byte;
371 3     rangev1=bytevar;
372 3     call incrmacroptr;
373 3     if range=doublerange then$do
374 4       rangev2=bytevar;
375 4       call incrmacroptr;
376 4     endif;
377 4     opvalue=oper.offset;
378 3     if range=doublerange then$do
381 4       return ((opvalue>=rangev1) and (opvalue<=rangev2));
382 4     else$do
384 4       return (opvalue=rangev1);
385 4     endif;
386 3   end rangetest;
```

```
387 2   modlettertest: proc byte;
388 3     dcl numb addr,(styp,modbyt) byte;
389 3     styp=oper.stype;
390 3     if styp=lab then return true;
392 3     modbyt=oper.sflag and typebit;
393 3     if styp = reg then return (modbyt=modletter);
395 3     if styp = variable then
396 4       return ((modbyt=nomod) or (modbyt=modletter));
397 3     if styp=number then$do
399 4       numb=oper.offset;
400 4       do case modletter-1;
401 5         return not wrdtest(numb); /* BYTE */
402 5         return wrdtest(numb); /* WORD */
403 5         return (typecalc(numb)=byt); /* signed BYTE */
404 5         return false; /* DWORD */
405 5       end$case;
406 4     endif;
407 3     return false;
408 3   end modlettertest;
```

```
409 2   speclettertest: proc byte;
410 3     dcl (opertype,locvalue,loctype) byte;
411 3     memtest: proc byte;
412 4       return (opertype=variable);
413 4     end memtest;
```

```
414 3   opertype=oper.stype;
415 3   locvalue=oper.offset;
416 3   loctype=oper.sflag and typebit;
417 3   do case specletter;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

```

418 4      /* A - accumulator (AX or AL) */
         return ((opertype=reg) and (locvalue=eax));
419 4      /* C - code reference, i.e. label */
         return (opertype=label);
420 4      /* D - immediate data */
         return (opertype=number);
421 4      /* E - effective address, i.e. memory address or register */
         return (memtest or (opertype=reg));
422 4      /* M - memory address */
         return memtest;
423 4      /* R - register except segment register */
         return ((opertype=reg) and (loctype <> dword));
424 4      /* S - segment register */
         return ((opertype=reg) and (loctype = dword));
425 4      /* X - memory address without indexing */
         return ((opertype=variable) and
426 4             ((oper.sflag and (iregbit or bragbit))=0));
427 3      end$case;
         end speclettertest;

428 2      specletter=bytevar; /* pick up codemacro attributes */
429 2      call incrmacroptr;
430 2      modletter=bytevar and modletter$bit;
431 2      range=bytevar and range$and;
432 2      rangetype=bytevar and rangetype$and;
433 2      call incrmacroptr;
434 2      pt=.operands(opno); /* address of current user operand */

435 2      match=true;
436 2      if range <> norange then match=rangetest;
438 2      if modletter <> 0 then match=match and modlettertest;
440 2      if match then match=speclettertest;
442 2      return match;
443 2      end matchsingleop;

/* test if operands match a specific codemacro */
444 1      matchingops: proc byte;
445 2      dcl savept address,(nopara,match,parno) byte;
446 2      savept=macroptr;
447 2      call incrmacroptr; /* macroptr=macroptr+2 */
448 2      call incrmacroptr;
449 2      nopara=bytevar; /* pick up no of parameters */
450 2      call incrmacroptr; /* advance to first formal */
451 2      if (nopara and prefix$on) <> 0 then return true; /* PREFIX */
453 2      if nopara <> nooper then$do
455 3          match=false;
456 3      else$do
458 3          match=true;
459 3          parno=0fffh;
460 3          do while (parno:=parno+1) < nopara;
461 4              match=match and matchsingleop(parno);
462 4          end$while;
463 3      end$if;
464 2      if not match then macroptr=savept;
466 2      return match;
467 2      end matchingops;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
468 1      /* test if operands match instruction */
469 2      searchformatch: proc byte public;
470 2          ucl next based macroptr address;
471 2          macroptr=firstmacroptr;
472 2          do forever;
473 3              if matchingops then return true;
474 3              if next=0 then return false;
475 3              macroptr=next;
476 3          end$forever;
477 2      and searchformatch;
478 2
479 1      end$module cmsubr;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
89	0000H	4	ABSADDR. BYTE ARRAY(4) EXTERNAL(25) 191 183 192
2			194
			LITERALLY 32 35 38 41 47 65 68
			71 74 89 90 120 150 160 165 173 187
			198 210 243 317 346 368 383
90	0000H	2	ADDRVAR. WORD BASED(MACROPTR) 155
55	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(14) STACK=0000H
58	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(15) STACK=0000H
243	007FH	1	BASEIND. BYTE 264 282 283 285
160	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
198	0008H	1	BASEINDEX. BYTE MEMBER(OPR) 202
346	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
165	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
173	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
210	0008H	1	BASEINDEX. BYTE MEMBER(OPR) 212 217
89	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
187	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
150	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
317	0008H	1	BASEINDEX. BYTE MEMBER(OPR)
243	0008H	1	BASEINDEX. BYTE MEMBER(OPR) 264
368	0008H	1	BASEINDEX. BYTE MEMBER(OPER)
88			BASEREGBIT LITERALLY 264 282 285
88			BASEREGCOUNT LITERALLY 212
90	0000H	2	BITCOUNT3. BYTE ARRAY(2) DATA 357
323	0006H	8	BITTAB BYTE ARRAY(8) DATA 332
88			BREGBIT. LITERALLY 213 254 272 425
88			BREGCOUNT. LITERALLY
4			BYT. LITERALLY 403
120	0004H	1	BYT1 BYTE AT AUTOMATIC 121
120	0005H	1	BYT2 BYTE AT AUTOMATIC 122
90	0000H	1	BYTEVAR. BYTE BASED(MACROPTR) 129 146 199 312 340
			342 347 351 356 357 371 375 428 430 431
			432 449
111	0004H	1	CH BYTE PARAMETER AUTOMATIC 112 115
80	0000H	1	CH BYTE PARAMETER 81
49	0000H	1	CH BYTE PARAMETER 50
52	0000H	1	CH BYTE PARAMETER 53
58	0000H	1	CH BYTE PARAMETER 59
55	0000H	1	CH BYTE PARAMETER 56
61	0000H	1	CH BYTE PARAMETER 62
134	0000H	1	CH BYTE BASED(PT) ARRAY(1) 137
46	0000H	1	CH BYTE PARAMETER 47
89	0000H	2	CIP. WORD EXTERNAL(26) 175 191
91	0000H	10	CLEARCINDEX PROCEDURE PUBLIC STACK=0002H
1	0000H		CMSURR PROCEDURE STACK=0000H
3			CODE LITERALLY
133	00C3H	50	COMMANDTYPE. PROCEDURE BYTE PUBLIC STACK=0000H 357
133	0008H	1	COMNO. BYTE PARAMETER AUTOMATIC 134 137
37	0000H		COPY PROCEDURE EXTERNAL(8) STACK=0000H

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		CR	LITERALLY	
323	0083H	1 CRSTI	BYTE	330 332 333 335
16		CSDATA	LITERALLY	98 103
89	0000H	2 CSLOVALUE	WORD EXTERNAL(27)	176 189
11		CSDVER	LITERALLY	227
16		CSVALUE	LITERALLY	
37	0000H	2 D	WORD PARAMETER	33
73	0000H	2 D	WORD PARAMETER	74
70	0000H	2 D	WORD PARAMETER	71
67	0000H	2 D	WORD PARAMETER	68
64	0000H	2 D	WORD PARAMETER	65
2		DCL	LITERALLY	
73	0000H	DECOU	PROCEDURE EXTERNAL(20) STACK=0000H	
3		DELETEDSYM	LITERALLY	
49	0000H	DIGIT	PROCEDURE BYTE EXTERNAL(12) STACK=0000H	
243	0078H	1 DISPLAG	BYTE	269 276 279 280 288 290 303 306
243	0013H	1 DISPHIGH	BYTE AT	307
173	000AH	2 DISPL	WORD	175 176 180
243	0012H	1 DISPLow	BYTE AT	305
244	03F0H	34 DISPTYPE	PROCEDURE BYTE STACK=0006H	279
3		DOUBLEDEFINED	LITERALLY	
15		DOUBLERANGE	LITERALLY	373 379
16		DRCODEDATA	LITERALLY	
16		DRCODESEGM	LITERALLY	
16		ORDATADATA	LITERALLY	
16		ORDATASEGM	LITERALLY	
16		DREXTRADATA	LITERALLY	
16		DREXTRASEGM	LITERALLY	
16		DRSTACKDATA	LITERALLY	
16		DRSTACKSEGM	LITERALLY	
16		DSDATA	LITERALLY	
11		DSOVER	LITERALLY	235
211	02CDH	45 DSOVERTEST	PROCEDURE BYTE STACK=0004H	230 234
16		DSVALUE	LITERALLY	
4		DWRD	LITERALLY	423 424
2		ELSEDO	LITERALLY	291
94	000AH	40 EMIT	PROCEDURE PUBLIC STACK=0008H	
90	001EH	80 EMITBYTE	BYTE ARRAY(80)	98 115
80	0000H	EMITCODEBYTE	PROCEDURE EXTERNAL(23) STACK=0000H	93 108
101	0032H	56 EMITDUMMIES	PROCEDURE PUBLIC STACK=0008H	
90	006EH	1 EMITINDEX	BYTE	92 97 115 116
76	0000H	EMITINIT	PROCEDURE EXTERNAL(21) STACK=0000H	
111	006AH	31 EMITSINGLEBYTE	PROCEDURE STACK=0004H	121 122 146 152
			180 239 302 305 307 365	
119	0089H	19 EMITSINGLEWORD	PROCEDURE STACK=000AH	155 162 169 170
			191	
78	0000H	EMITTERMINATE	PROCEDURE EXTERNAL(22) STACK=0000H	
29	0000H	EMPTYLINE	PROCEDURE BYTE EXTERNAL(5) STACK=0000H	
2		ENDCASE	LITERALLY	
2		ENDDO	LITERALLY	
2		ENDFOREVER	LITERALLY	
2		ENDIF	LITERALLY	239 307
2		ENDMODULE	LITERALLY	
2		ENDOFFILE	LITERALLY	
2		ENDPROC	LITERALLY	
2		ENDWHILE	LITERALLY	141
16		EOTYPE	LITERALLY	

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

64	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(17) STACK=0000H		
83	0000H		ERRMSG.	PROCEDURE EXTERNAL(24) STACK=0000H	168	179
				190		
83	0000H	1	ERRNO.	BYTE PARAMETER	84	
3			ERROR.	LITERALLY		
16			ESDATA.	LITERALLY		
11			ESOVER.	LITERALLY	223	
16			ESVALUE.	LITERALLY		
2			FALSE.	LITERALLY	404 407 455 475	
43	0000H	2	FCBADR.	WORD PARAMETER	44	
43	0000H		FILEABORT.	PROCEDURE EXTERNAL(10) STACK=0000H		
46	0000H		FILL.	PROCEDURE EXTERNAL(11) STACK=0000H		
89	0000H	2	FIRSTMACROPTR.	WORD EXTERNAL(29)	470	
198	0074H	1	FLAG.	BYTE	204 205	
2			FOREVER.	LITERALLY	471	
345	0536H	53	FORMATTSROUT.	PROCEDURE STACK=0000H		359
2			FORMFEED.	LITERALLY		
89	0000H	1	FULLSYMTAB.	BYTE EXTERNAL(31)		
127	00A5H	30	GETOPERADR.	PROCEDURE WORD STACK=0006H	151 161 166	
				174 188 201 215 314 318 320 349		
67	0000H		HEX1OUT.	PROCEDURE EXTERNAL(18) STACK=0000H		
70	0000H		HEX2OUT.	PROCEDURE EXTERNAL(19) STACK=0000H	183	194
134	0072H	1	I.	BYTE	135 136 137 140	
102	0070H	1	I.	BYTE	106 107	
95	006FH	1	I.	BYTE	96 97 98	
3			IDENT.	LITERALLY		
124	009CH	9	INCRMACROPTR.	PROCEDURE STACK=0002H	130 139 147 156	
				157 200 313 341 343 348 352 364 372 376		
				429 433 447 448 450		
88			INDEXREGBIT.	LITERALLY	264 282 283	
88			INDEXREGCOUNT.	LITERALLY		
251	0412H	29	INDEXTYPE.	PROCEDURE BYTE STACK=0002H	281	
2			INIT.	LITERALLY		
16			INTELDATA.	LITERALLY		
16			INTELSEGMENT.	LITERALLY		
88			IREGBIT.	LITERALLY	252 272 425	
88			IREGCOUNT.	LITERALLY		
102	0071H	1	J.	BYTE	103 104 105 107	
324	04B6H	92	JOIN.	PROCEDURE STACK=000AH	342 351	
3			LAB.	LITERALLY	390 419	
86			LABOUTOFRANGE.	LITERALLY	178 190	
7			LEFTBRACKET.	LITERALLY		
150	0000H	2	LENGTH.	WORD MEMBER(COPR)		
346	0000H	2	LENGTH.	WORD MEMBER(COPR)		
160	0000H	2	LENGTH.	WORD MEMBER(COPR)		
210	0000H	2	LENGTH.	WORD MEMBER(COPR)		
165	0000H	2	LENGTH.	WORD MEMBER(COPR)		
317	0000H	2	LENGTH.	WORD MEMBER(COPR)		
187	0000H	2	LENGTH.	WORD MEMBER(COPR)		
173	0000H	2	LENGTH.	WORD MEMBER(COPR)		
198	0000H	2	LENGTH.	WORD MEMBER(COPR)		
368	0000H	2	LENGTH.	WORD MEMBER(COPR)		
243	0000H	2	LENGTH.	WORD MEMBER(COPR)		
			LENGTH.	BUILTIN	357	
89	0000H	2	LENGTH.	WORD MEMBER(COPERANDS)		
52	0000H		LETTER.	PROCEDURE BYTE EXTERNAL(13) STACK=0000H		
2			LF.	LITERALLY		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

133	0006H	1	LG	BYTE PARAMETER AUTOMATIC	134	135	143
2			LIT.	LITERALLY	2	3	4
					9	11	12
					13	14	15
					16	17	18
					19	20	21
410	0093H	1	LOC TYPE	BYTE	416	423	424
410	0092H	1	LOC VALUE	BYTE	415	418	
89	0000H	2	MACROPTR	WORD EXTERNAL(30)	90	125	129
					139	312	340
					342	347	351
					356	357	371
					428	430	431
					432	446	449
					465	469	470
					474		
368	0087H	1	MATCH.	BYTE	435	437	439
445	0095H	1	MATCH.	BYTE	455	458	461
444	07AEH	108	MATCHINGOPS.	PROCEDURE	BYTE STACK=0010H		472
367	0568H	111	MATCHSINGLEOP.	PROCEDURE	BYTE STACK=000CH		451
8			MODF	LITERALLY			
149	0106H	21	MODFROUT	PROCEDURE	PUBLIC STACK=000AH		
8			MODIT.	LITERALLY			
322	046BH	75	MODITROUT.	PROCEDURE	PUBLIC STACK=0012H		
8			MODN	LITERALLY			
145	00F5H	17	MODNROUT	PROCEDURE	PUBLIC STACK=0008H		
8			MODDF	LITERALLY			
164	0144H	47	MODDFROUT	PROCEDURE	PUBLIC STACK=000EH		
8			MODWF	LITERALLY			
159	012FH	21	MODWFROUT	PROCEDURE	PUBLIC STACK=000EH		
8			MODWN	LITERALLY			
154	011BH	20	MODWNROUT	PROCEDURE	PUBLIC STACK=000EH		
411	079FH	15	MEMTEST.	PROCEDURE	BYTE STACK=0002H		421
8			MENBIT.	LITERALLY	356		422
8			MENDM.	LITERALLY			
8			MFORMALBITS.	LITERALLY	90		
86			MISSEGINFN.	LITERALLY	168		
8			MMODRM1.	LITERALLY			
310	042FH	28	MMODRM1ROUT.	PROCEDURE	PUBLIC STACK=0012H		
8			MMODRM2.	LITERALLY			
316	044BH	32	MMODRM2ROUT.	PROCEDURE	PUBLIC STACK=0012H		
8			MNOSEGFIX.	LITERALLY			
197	0218H	60	MNOSEGFIXROUT.	PROCEDURE	PUBLIC STACK=000AH		
8			MNUMBERBITS.	LITERALLY	90		
10			MODB	LITERALLY			
388	0090H	1	MODBYT	BYTE	392	394	396
10			MODD	LITERALLY			
243	0079H	1	MODFTELD	BYTE	268	275	280
368	0089H	1	MODLETTER.	BYTE	394	396	400
14			MODLETTERBIT	LITERALLY	430		438
14			MODLETTERCOUNT	LITERALLY			
387	063AH	162	MODLETTERTEST.	PROCEDURE	BYTE STACK=0006H		439
242	02FAH	246	MODRM.	PROCEDURE	STACK=000EH	314	320
10			MODSB.	LITERALLY			
10			MODW	LITERALLY			
8			MRELB.	LITERALLY			
172	0173H	96	MRELBROUT.	PROCEDURE	PUBLIC STACK=000AH		
8			MRELW.	LITERALLY			
186	01D3H	69	MRELWROUT.	PROCEDURE	PUBLIC STACK=000EH		
8			MSEGFIX.	LITERALLY			
209	0254H	121	MSEGFIXROUT.	PROCEDURE	PUBLIC STACK=000AH		
73	0000H	2	N.	WORD PARAMETER	74		
70	0000H	2	N.	WORD PARAMETER	71		

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

67	0000H	1	N.	BYTE PARAMETER	53			
64	0000H	1	N.	BYTE PARAMETER	65			
46	0000H	1	N.	BYTE PARAMETER	47			
34	0000H	2	N.	WORD PARAMETER	35			
37	0000H	1	N.	BYTE PARAMETER	33			
3			NEGLECTED.	LITERALLY				
469	0000H	2	NEXT	WORD BASED(MACROPTR)	474	476		
4			NIL.	LITERALLY				
339	0084H	1	NOBIT.	BYTE 340 342				
346	0089H	1	NOBIT.	BYTE 347 351				
324	0006H	1	NOBIT.	BYTE PARAMETER AUTOMATIC	325	328	329	330
				334				
102	0002H	4	NDDUM.	BYTE ARRAY(4) DATA	107			
89	0000H	1	NOERRRK.	BYTE EXTERNAL(23)	113	206		
10			NOMOD.	LITERALLY 396				
89	0000H	1	NOOPER.	BYTE EXTERNAL(32)	103	453		
88			NOOVERCOUNT.	LITERALLY				
88			NOOVERKTOEBIT.	LITERALLY 202 217				
445	0094H	1	NOPARA.	BYTE 449 451 453 460				
12			NOPARAND.	LITERALLY				
15			NORANGE.	LITERALLY 436				
324	0004H	1	NOSHIFT.	BYTE PARAMETER AUTOMATIC	325	326	327	
346	0086H	1	NUMD.	BYTE 350 351				
324	0008H	1	NUMB.	BYTE PARAMETER AUTOMATIC	325	327	329	331
				335				
388	001AH	2	NUMB.	WORD 399 401 402 403				
3			NUMBER.	LITERALLY 397 420				
338	0512H	36	NUMBERBITSR0UT.	PROCEDURE STACK=000FH	358			
15			NUMBERRANGE.	LITERALLY				
7			OAND.	LITERALLY				
7			DEQ.	LITERALLY				
346	0006H	2	OFFSET.	WORD MEMBER(OPR)	350			
317	0006H	2	OFFSET.	WORD MEMBER(OPR)	319			
243	0006H	2	OFFSET.	WORD MEMBER(OPR)	260			
198	0006H	2	OFFSET.	WORD MEMBER(OPR)				
187	0006H	2	OFFSET.	WORD MEMBER(OPR)	191	194		
173	0006H	2	OFFSET.	WORD MEMBER(OPR)	175	183		
165	0006H	2	OFFSET.	WORD MEMBER(OPR)	169			
160	0006H	2	OFFSET.	WORD MEMBER(OPR)	162			
150	0006H	2	OFFSET.	WORD MEMBER(OPR)	152			
210	0006H	2	OFFSET.	WORD MEMBER(OPR)				
89	0006H	2	OFFSET.	WORD MEMBER(OPERANDS)				
243	0012H	2	OFFSET.	WORD 243 247 249	260	267		
368	0006H	2	OFFSET.	WORD MEMBER(OPER)	378	399	415	
7			UGE.	LITERALLY				
7			OGT.	LITERALLY				
7			OLAST.	LITERALLY				
7			OLE.	LITERALLY				
7			OLENGTH.	LITERALLY				
7			OLT.	LITERALLY				
7			OMOD.	LITERALLY				
7			ONE.	LITERALLY				
7			ONOT.	LITERALLY				
7			OFFSET.	LITERALLY				
7			ODR.	LITERALLY				
368	0000H	9	OPER.	STRUCTURE BASED(PT)	378	389	392	399 414
				415 416 425				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

89	0000H	36	OPERANDS	STRUCTURE	ARNSYC4) EXTERNALC33)	127	434
87			OPERANDSFRUC	LITERALLY	30 150 160 165 173 187 194		
					210 243 317 346 358		
3			OPERATOR	LITERALLY			
410	0091H	1	OPERTYPE	BYTE	412 414 418 419 420 421 423 424		
					425		
370	0082H	1	OPERVALUE	BYTE	378 381 384		
367	0004H	1	OPNO	BYTE	PARAMETER AUTOMATIC	368 434	
346	0000H	9	OPR	STRUCTURE	BASED(PT)	350	
317	0000H	9	OPR	STRUCTURE	BASED(PT)	319	
210	0000H	9	OPR	STRUCTURE	BASED(PT)	212 216 217	
198	0000H	9	OPR	STRUCTURE	BASED(PT)	202 204	
187	0000H	9	OPR	STRUCTURE	BASED(PT)	189 191 194	
173	0000H	9	OPR	STRUCTURE	BASED(PT)	175 176 183	
165	0000H	9	OPR	STRUCTURE	BASED(PT)	167 169 170	
160	0000H	9	OPR	STRUCTURE	BASED(PT)	162	
150	0000H	9	OPR	STRUCTURE	BASED(PT)	152	
243	0000H	9	OPR	STRUCTURE	BASED(PT)	260 261 262 264	
7			OPTR	LITERALLY			
7			OSEG	LITERALLY			
7			OSHL	LITERALLY			
7			OSHORT	LITERALLY			
7			OSHR	LITERALLY			
7			OTYPE	LITERALLY			
40	0000H		OUTTEXT	PROCEDURE	EXTERNAL(9) STACK=0000H		
210	0077H	1	OVERRIDE	BYTE	222 226 230 234 238		
7			OXOR	LITERALLY			
445	0096H	1	PARNO	BYTE	459 460 461		
6			PCODEMACRO	LITERALLY			
6			PCSEG	LITERALLY			
6			PDB	LITERALLY			
6			PDBIT	LITERALLY			
6			PDD	LITERALLY			
6			PDSEG	LITERALLY			
6			PDW	LITERALLY			
6			PEJECT	LITERALLY			
6			PEND	LITERALLY			
6			PENDIF	LITERALLY			
6			PENDM	LITERALLY			
6			PEQU	LITERALLY			
6			PESEG	LITERALLY			
6			PIF	LITERALLY			
6			PIFLIST	LITERALLY			
6			PINCLUDE	LITERALLY			
6			PLIST	LITERALLY			
6			PHODRM	LITERALLY			
6			PNOIFLIST	LITERALLY			
6			PNOLIST	LITERALLY			
6			PNOSEGFIX	LITERALLY			
6			PORG	LITERALLY			
6			PPAGESIZE	LITERALLY			
6			PPAGENIDTH	LITERALLY			
6			PRB	LITERALLY			
12			PREFIXAND	LITERALLY			
13			PREFIXDN	LITERALLY			
6			PRELB	LITERALLY			
6			PRELW	LITERALLY			

CCP/M-86 2.0 V.3
 COPYR.GHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

2		PROC	LITERALLY	17	19	21	23	25	29	31
			34 55 53 67 79 75 76 75 50 33							
			91 94 101 111 119 124 127 133 145 149							
			154 159 164 172 186 197 209 211 242 244							
			251 310 316 322 324 338 345 357 369 387							
			409 411 444 463							
6		PRS.	LITERALLY							
6		PRW.	LITERALLY							
6		PSEGFTX.	LITERALLY							
3		PSEUDO	LITERALLY							
6		PSIMFORM	LITERALLY							
6		PSSEG.	LITERALLY							
346	0016H	2 PT	WORD	346	349	350				
317	0014H	2 PT	WORD	317	318	319				
210	0010H	2 PT	WORD	210	212	215	216	217		
198	000EH	2 PT	WORD	198	201	202	204			
187	000CH	2 PT	WORD	187	188	189	191	194		
173	0008H	2 PT	WORD	173	174	175	176	183		
165	0006H	2 PT	WORD	165	166	167	169	170		
160	0004H	2 PT	WORD	160	161	162				
150	0002H	2 PT	WORD	150	151	152				
133	0004H	2 PT	WORD PARAMETER	AUTOMATIC			134	137		
128	0000H	2 PT	WORD	129	131					
242	0004H	2 PT	WORD PARAMETER	AUTOMATIC			243	260	261	262
			264							
46	0000H	2 PT	WORD PARAMETER	47						
368	0018H	2 PT	WORD	368	378	389	392	399	414	415 416
			425 434							
6		PTITLE	LITERALLY							
5		RAH.	LITERALLY							
5		RAL.	LITERALLY							
368	008AH	1 RANGE.	BYTE	373	379	431	436			
15		RANGEAND	LITERALLY	431						
14		RANGESPECBIT	LITERALLY							
14		RANGESPECCOUNT	LITERALLY							
369	05DAH	96 RANGETEST.	PROCEDURE	BYTE	STACK=0006H			437		
368	008BH	1 RANGETYPE.	BYTE	432						
15		RANGETYPEAND	LITERALLY	432						
368	008CH	1 RANGEV1.	BYTE	371	381	384				
368	008DH	1 RANGEV2.	BYTE	375	381					
5		RAX.	LITERALLY	418						
5		RBH.	LITERALLY							
5		RBL.	LITERALLY							
5		RBP.	LITERALLY							
5		RBX.	LITERALLY							
5		RCH.	LITERALLY							
5		RCL.	LITERALLY							
5		RCS.	LITERALLY	245						
5		RCX.	LITERALLY							
5		RDH.	LITERALLY							
5		RDI.	LITERALLY							
5		RDL.	LITERALLY							
5		RDS.	LITERALLY							
5		RDX.	LITERALLY							
2		REENT.	LITERALLY							
3		REG.	LITERALLY							
317	0081H	1 REGFIELD	BYTE	319	320	265	393	418	421	423 424

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

311	0080H	1	REGFIELD	BYTE	312	314													
242	0006H	1	REGFIELD	BYTE	PARAMETER	AUTOMATIC	243	300	302										
15			REGISTER RANGE	LITERALLY															
5			RES.	LITERALLY															
323	0082H	1	RESULT	BYTE	332	354	365												
7			RIGHTBRACKET	LITERALLY															
243	007AH	1	RMFIELD.	BYTE	267	274	282	283	287	294	302								
5			RSI.	LITERALLY															
5			RSP.	LITERALLY															
5			RSS.	LITERALLY															
64	0000H	2	S.	WORD	PARAMETER		65												
37	0000H	2	S.	WORD	PARAMETER		38												
445	001CH	2	SAVEPT	WORD	446	465													
19	0000H		SCAN	PROCEDURE	EXTERNAL(1)	STACK=0000H													
17	0000H		SCANINIT	PROCEDURE	EXTERNAL(0)	STACK=0000H													
468	081AH	43	SEARCHFORMATCH	PROCEDURE	BYTE	PUBLIC	STACK=0014H												
88			SEGMBIT.	LITERALLY	167														
88			SEGMCOUNT.	LITERALLY															
368	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
346	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
317	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
243	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
210	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
198	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
187	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)		189												
173	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)		176												
165	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)		170												
160	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
150	0004H	2	SEGMENT.	WORD	MEMBER(COPIR)														
89	0004H	2	SEGMENT.	WORD	MEMBER(COPIR ANDS)														
243	007EH	1	SEGR	BYTE	245	263													
210	0076H	1	SEGR	BYTE	212	213	219	220	223	227	231	235							
					239														
198	0073H	1	SEGR	BYTE	199	206													
198	0075H	1	SEGT	BYTE	205	206													
88			SEGTYP=BIT	LITERALLY															
88			SEGTYP=COUNT	LITERALLY	205	219	263												
368	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)		392	416	425										
346	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)														
317	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)														
243	0070H	1	SFLAG.	BYTE	252	254	262	263	272										
243	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)		262												
210	0078H	1	SFLAG.	BYTE	213	216	219												
210	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)		216												
198	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)		204												
187	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)														
173	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)														
165	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)		167												
160	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)														
150	0003H	1	SFLAG.	BYTE	MEMBER(COPIR)														
89	0003H	1	SFLAG.	BYTE	MEMBER(COPIR ANDS)														
			SHL.	BUILTIN	300	301	329	335											
			SHR.	BUILTIN	205	212	219	263	327										
15			SINGLE RANGE.	LITERALLY															
21	0000H		SKIPRESTOFFLINE	PROCEDURE	EXTERNAL(2)	STACK=0000H													
26	0000H		SKIPUNTIL.	PROCEDURE	BYTE	EXTERNAL(4)	STACK=0000H												
2			SPACE.	LITERALLY	181	192													

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

Address	Hex	Label	Symbol	Value	Address	Hex	Label	Symbol	Value	Address	Hex	Label	Symbol	Value	Address	Hex	Label	Symbol	Value	
3			SPEC					LITERALLY												
9			SPECA					LITERALLY												
9			SPECC					LITERALLY												
9			SPECD					LITERALLY												
9			SPECE					LITERALLY												
23	0000H		SPECIALTOKLN					PROCEDURE	BYTE	EXTERNAL(3)	STACK=0000H									
368	008BH	1	SPECLITER					BYTE	417	428										
409	06DCH	195	SPECLITERTEST					PROCEDURE	BYTE	STACK=0006H										441
9			SPECM					LITERALLY												
9			SPECR					LITERALLY												
9			SPECS					LITERALLY												
9			SPECX					LITERALLY												
16			SSDATA					LITERALLY												
11			SSOVER					LITERALLY		231										
16			SSVALUE					LITERALLY												
16			STARTTYPE					LITERALLY												
3			STKING					LITERALLY												
2			STRUC					LITERALLY												
										89	150	160	165	173	187	198				
									210	243	317	346	368							
388	008FH	1	STYP					BYTE	389	390	393	395	397							
368	0002H	1	STYPE					BYTE	MEMBER(CUPR)			389	414							
346	0002H	1	STYPE					BYTE	MEMBER(CUPR)											
317	0002H	1	STYPE					BYTE	MEMBER(CUPR)											
243	007CH	1	STYPE					BYTE	261	265										
243	0002H	1	STYPE					BYTE	MEMBER(CUPR)			261								
210	0002H	1	STYPE					BYTE	MEMBER(CUPR)											
198	0002H	1	STYPE					BYTE	MEMBER(CUPR)											
187	0002H	1	STYPE					BYTE	MEMBER(CUPR)											
173	0002H	1	STYPE					BYTE	MEMBER(CUPR)											
165	0002H	1	STYPE																	

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. #

402

MODULE INFORMATION:

CODE AREA SIZE = 0345H 2117D
 CONSTANT AREA SIZE = 000EH 14D
 VARIABLE AREA SIZE = 0097H 151D
 MAXIMUM STACK SIZE = 0014H 20D
 933 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____



IAS-11 PL/M-86 V2.0 COMPILATION OF MODULE INSTRU

OBJECT MODULE PLACED IN INSTR.OPJ

COMPILER INVOKED BY: :F0: INSTR.PLH DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

1

{title ("INSTRUCTION MODULE")

instruc:

do:

/*

This is the module to decode and produce code-
output of a single instruction, possibly preceded
by a number of PREFIX-instructions.

*/

```
$include (:f1:macro.lit)
= $nolist
$include (:f1:struc.lit)
= $nolist
$include (:f1:cmacd.lit)
= $nolist
$include (:f1:equals.lit)
= $nolist
$include (:f1:ermod.lit)
= $nolist
$include (:f1:subr1.ext)
= $nolist
$include (:f1:expr.ext)
= $nolist
$include (:f1:symb.ext)
= $nolist
$include (:f1:scan.ext)
= $nolist
$include (:f1:ermod.ext)
= $nolist
$include (:f1:cmsubr.ext)
= $nolist
$include (:f1:instr.x86)
= $nolist
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
120 1  reject
      decl /* global variables */
      bytevar based macroptr byte, /* byte within code macro */
      contab(12) byte data /* legal code macro commands */
      (mdbn,mdbf,mown,adwf,mddf,mrelb,mrelw,amodrm1,amodrm2,msegfix,
       mnosegfix,mdbit);
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$eject

```

121 1  /* generate instruction output code */
122 2  makecode: proc byte;
      if (noerror:=searchformat) then$do
        /* matching operands, compute code */
124 3  do while (bytevar <> mandm) and noerror;
125 4    do case commandtype(bytevar,length(comtab),.comtab);
126 5      call mDBItrout;
127 5      call mDBFrout;
128 5      call mDWItrout;
129 5      call mDWFrout; /* typed during earthquake */
130 5      call mDDFrout;
131 5      call mRELPTrout;
132 5      call mkELWTrout;
133 5      call mMODRM1rout;
134 5      call mMODRM2rout;
135 5      call mSEGFIXrout;
136 5      call mMOSEGFIXrout;
137 5      call mDBITrout;
138 5      do; end; /* dummy, should not happen */
140 5    end$case;
141 4  end$while;
142 3  end$if;
143 2  if noerror then call emit; else call emitdummies;
146 2  return noerror;
147 2  end makecode;

148 1  /* scan all PREFIX instructions */
      prefixscan: proc byte;
        /* compute address of first codemacro */
149 2  findmacroaddr: proc;
150 3    dcl macrop based codemacroptr address;
151 3    firstmacroptr=macrop;
152 3  end findmacroaddr;
      /* test if instruction is of PREFIX type */
153 2  prefixinstr: proc byte;
154 3    dcl ptr address,flag based ptr byte;
155 3    ptr=firstmacroptr+2;
156 3    return ((flag and prefix$on) <> 0);
157 3  end prefixinstr;

158 2  call findmacroaddr; /* compute pointer to first macro */
159 2  do while prefixinstr;
160 3    if makecode then; /* generate output code,always succed */
162 3      call clearcindex;
163 3      if findcodemacro(acclen,.accum(0),.codemacroptr) then$do
165 4        call scan;
166 4        call findmacroaddr;
167 4      else$do
169 4        call errmsy(missinstr); /* missing instruction */
170 4        call skip$rest$of$line;
171 4        return false;
172 4      end$if;
173 3    end$while;
174 2  return true;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

175 2      end prefixscan;

/* get all instruction operands */
176 1      getoperands: proc byte;
177 2      dcl moreoperands byte,pt address,oper based pt operandstruc,
          exitvalue byte;

178 2      exitvalue=true;
179 2      nooper=0; /* clear no of operands */
180 2      moreoperands=not emptyline;
181 2      do while moreoperands;
182 3      moreoperands=false;
183 3      pt=.operands(nooper);
184 3      if not operand then%do
186 4      if oper.stype <> udefsyms then call errmsg(illiooper);
188 4      exitvalue=false;
189 4      if skip$until(",") then moreoperands=true;
          else%do
193 4      if specialtoken(",") then%do
195 5      call scan; /* skip ",", " */
196 5      if nooper < 3 then moreoperands=true;
          end%if;
          end%if;
200 3      nooper=nooper+1;
201 3      end$while;
202 2      return exitvalue;
203 2      end getoperands;

/* test if operands contain enough type information */

204 1      enough$type$info: proc byte;
205 2      dcl pt address,oper based pt operandstruc,(i,flag) byte;
206 2      flag=true;
207 2      i=0ffh;
208 2      do while (i:=i+1) < nooper;
209 3      pt=.operands(i);
210 3      if oper.stype=variable then%do
212 4      if (oper.sflag and typebit) = 0 then flag=false;
          end%if;
          end$while;
216 2      if flag then return true;
218 2      i=0ffh; /* one of operands lacks type info,check others */
219 2      do while (i:=i+1) < nooper;
220 3      pt=.operands(i);
221 3      if (oper.sflag and typebit) <> 0 then return true;
223 3      if (oper.stype=number) and (wrctest(oper.offset)) then return true;
          end$while;
226 2      return false;
227 2      end enough$type$info;

/* Module entry point: */

228 1      instruction: proc public; /* decode line in pass 1 and pass 2 */
229 2      call clearcmindex; /* clear buffer for output codes */
230 2      if prefixscan then%do
232 3      if getoperands then%do

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
234 4      if enough$type$info then$do
236 5          if makecode then$do
238 6              if not emptyline then$do
240 7                  call errmsg(end$of$line$err);
241 7                  end$if;
                else$do
244 6                    call errmsg(opaismatch);
245 6                    end$if;
246 5                else$do
248 5                    call errmsg(misstypeinfo);
249 5                    call emitdummies;
250 5                    end$if;
251 4                else$do
253 4                    if makecode then; /* try to make code with bad operands */
                        end$if;
256 3                end$if;
257 2                call skip$rest$of$line;
258 2            end instruction;

259 1    end$module instruc;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
119	0000H	1	ACCLEM BYTE EXTERNAL(43) 163
119	0000H	30	ACCUM. BYTE ARRAY(80) EXTERNAL(44) 163
2			ADDR LITERALLY 19 22 25 37 38 44 47 50 53 56 59 62 65 119 177 205
64	0000H	2	ASCTPTR WORD PARAMETER 65
119	0008H	1	BASEINDEX. BYTE MEMBER(COPERANDS)
205	0008H	1	BASEINDEX. BYTE MEMBER(COPER)
177	0008H	1	BASEINDEX. BYTE MEMBER(COPER)
3			BASEREGBIT LITERALLY
3			BASEREGCOUNT LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
13			BYT. LITERALLY
120	0000H	1	BYTEVAR. BYTE BASED(MACROPTR) 124 125
84	0000H		CLEARCMTNDLX PROCEDURE EXTERNAL(26) STACK=0000H 162 229
12			CODE LITERALLY
17			CODEMACROERR LITERALLY
3			CODEMACROHEAD. LITERALLY
119	0000H	2	CODEMACROPTR WORD EXTERNAL(50) 150 151 163
90	0000H		COMMANDTYPE. PROCEDURE BYTE EXTERNAL(29) STACK=0000H 125
90	0000H	1	COMNO. BYTE PARAMETER 91
120	0000H	12	CONTR. BYTE ARRAY(12) DATA 125
24	0000H		COPY PROCEDURE EXTERNAL(2) STACK=0000H
2			CR LITERALLY
7			CSTACK LITERALLY
24	0000H	2	D. WORD PARAMETER 25
2			DCL. LITERALLY
12			DELETEDSYMB. LITERALLY
52	0000H	2	DEST WORD PARAMETER 53
12			DOUBLEDEFINED. LITERALLY
17			DOUBLEDEFLAB LITERALLY
17			DOUBLEDEFSYMB. LITERALLY
17			DOUBLEDEFVAR LITERALLY
11			DOUBLERANGE. LITERALLY
7			DSTACK LITERALLY
13			DWORD LITERALLY
2			ELSEDO LITERALLY 190 241
86	0000H		EMIT PROCEDURE EXTERNAL(27) STACK=0000H 144
88	0000H		EMITDUMMIES. PROCEDURE EXTERNAL(28) STACK=0000H 145 249
79	0000H		EMPTYLINE. PROCEDURE BYTE EXTERNAL(24) STACK=0000H 180 238
2			ENDCASE. LITERALLY
2			ENDDO. LITERALLY
2			ENDFOREVER LITERALLY
2			ENDTF. LITERALLY 197 198 213 254
2			ENDMODULE. LITERALLY
2			ENDOFFILE. LITERALLY
17			ENDOFFLINEERR LITERALLY 240
38	0000H	2	ENDOSYMBTAB WORD EXTERNAL(8)

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		ENDPROC.	LITERALLY	
2		ENDWHILE	LITERALLY	214 224
204 0180H	143	ENOUGHTYPEINFO	PROCEDURE BYTE STACK=0000H	234
55 0000H		ENTERATTRIBUTES.	PROCEDURE EXTERNAL(15) STACK=0000H	
81 0000H		ERRMSG	PROCEDURE EXTERNAL(25) STACK=0000H	169 187
			240 244 248	
81 0000H	1	ERRNO.	BYTE PARAMETER	82
12		ERRDR.	LITERALLY	
7		ESOVER	LITERALLY	
177 0007H	1	EXITVALUE.	BYTE	178 188 202
31 0000H		EXPRESSION	PROCEDURE BYTE EXTERNAL(5) STACK=0000H	
2		FALSE.	LITERALLY	171 182 188 213 226
3		FILEISIRUCFURE	LITERALLY	
3		FILEOSTRUCTURE	LITERALLY	
17		FILESYNERR.	LITERALLY	
58 0000H		FINDCODEMACRO.	PROCEDURE BYTE EXTERNAL(16) STACK=0000H	163
149 00E0H	14	FINDMACROADDR.	PROCEDURE STACK=0002H	158 166
49 0000H		FINDSYMBOL	PROCEDURE BYTE EXTERNAL(13) STACK=0000H	
17		FIRSTITEM.	LITERALLY	
119 0000H	2	FIRSTMACROPTR.	WORD EXTERNAL(46)	151 155
205 0009H	1	FLAG	BYTE	206 213 216
154 0000H	1	FLAG	BYTE BASED(PTR)	156
2		FOREVER.	LITERALLY	
2		FORMFEED	LITERALLY	
37 0000H	2	FREET	WORD EXTERNAL(7)	
52 0000H		GETATTRIBUTES.	PROCEDURE EXTERNAL(14) STACK=0000H	
176 0109H	125	GETOPERANDS.	PROCEDURE BYTE STACK=0006H	232
205 0008H	1	I.	BYTE	207 208 209 218 219 220
12		IDENT.	LITERALLY	
17		IFPARERR.	LITERALLY	
17		ILLEGALMACRO	LITERALLY	
17		ILLEGALPSEUDO.	LITERALLY	
17		ILLEXPRELEM.	LITERALLY	
17		ILLTOPER	LITERALLY	187
3		INDEXREGBIT.	LITERALLY	
3		INDEXREGCOUNT.	LITERALLY	
2		INIT	LITERALLY	
17		INSTREKR	LITERALLY	
1 0000H		INSTRUC.	PROCEDURE STACK=0000H	
228 0215H	74	INSTRUCTION.	PROCEDURE PUBLIC STACK=0012H	
3		IREGBIT.	LITERALLY	
3		IREGCOUNT.	LITERALLY	
12		LAB.	LITERALLY	
17		LABOUTOF RANGE.	LITERALLY	
16		LEFTBRACKET.	LITERALLY	
205 0000H	2	LENGTH	WORD MEMBER(OPER)	
		LENGTH	BUILTIN	125
119 0000H	2	LENGTH	WORD MEMBER(OPERANDS)	
177 0000H	2	LENGTH	WORD MEMBER(OPER)	
2		LF	LITERALLY	
90 0000H	1	LG	BYTE PARAMETER	91
46 0000H	1	LG	BYTE PARAMETER	47
43 0000H	1	LG	BYTE PARAMETER	44
64 0000H	1	LG	BYTE PARAMETER	65
58 0000H	1	LG	BYTE PARAMETER	59
49 0000H	1	LG	BYTE PARAMETER	50
61 0000H	1	LG	BYTE PARAMETER	62

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2	LIT.	LITERALLY	2	3	4	5	6	7	8
		9	10	11	12	13	14	15	16
46	0000H	2	MACDEEPT	WORD PARAMETER	47				
150	0000H	2	MACROP	WORD BASED(CODE MACROPTK)	151				
119	0000H	2	MACROPTR	WORD EXTERNAL(47)	120	124	125		
121	0000H	162	MAKECODE	PROCEDURE BYTE STACK=0000H			160	236	253
4			MODF	LITERALLY	120				
95	0000H		MODFROUT	PROCEDURE EXTERNAL(31) STACK=0000H				127	
4			MODBT	LITERALLY	120				
115	0000H		MODBTROUT	PROCEDURE EXTERNAL(41) STACK=0000H				137	
4			MODN	LITERALLY	120				
93	0000H		MODNRROUT	PROCEDURE EXTERNAL(30) STACK=0000H				126	
4			MODF	LITERALLY	120				
101	0000H		MODFROUT	PROCEDURE EXTERNAL(34) STACK=0000H				130	
4			MODWF	LITERALLY	120				
99	0000H		MODWFROUT	PROCEDURE EXTERNAL(33) STACK=0000H				129	
4			MODWN	LITERALLY	120				
97	0000H		MODWNROUT	PROCEDURE EXTERNAL(32) STACK=0000H				128	
4			MENDBT	LITERALLY					
4			MENDN	LITERALLY	124				
4			MFORMALBITS	LITERALLY					
17			MISSIFERR	LITERALLY					
17			MISSINGPSEUDO	LITERALLY					
17			MISSINSTR	LITERALLY	169				
17			MISSEGINF0	LITERALLY					
17			MISSTYPEINF0	LITERALLY	248				
4			MMODRM1	LITERALLY	120				
111	0000H		MMODRM1ROUT	PROCEDURE EXTERNAL(39) STACK=0000H				133	
4			MMODRM2	LITERALLY	120				
113	0000H		MMODRM2ROUT	PROCEDURE EXTERNAL(40) STACK=0000H				134	
4			MNOSEGFTX	LITERALLY	120				
107	0000H		MNOSEGFTXROUT	PROCEDURE EXTERNAL(37) STACK=0000H				136	
4			MNUMBERBITS	LITERALLY					
6			MODB	LITERALLY					
6			MODD	LITERALLY					
10			MODLETTERBIT	LITERALLY					
10			MODLETTERCOUNT	LITERALLY					
6			MODSB	LITERALLY					
6			MODW	LITERALLY					
177	0006H	1	MOREOPERANDS	BYTE	180	181	182	190	197
4			MRELB	LITERALLY	120				
103	0000H		MRELBROUT	PROCEDURE EXTERNAL(35) STACK=0000H				131	
4			MRELW	LITERALLY	120				
105	0000H		MRELWROUT	PROCEDURE EXTERNAL(36) STACK=0000H				132	
4			MSEGFTX	LITERALLY	120				
109	0000H		MSEGFTXROUT	PROCEDURE EXTERNAL(38) STACK=0000H				135	
24	0000H	1	N	BYTE PARAMETER	25				
21	0000H	2	N	WORD PARAMETER	22				
12			NEGLECTED	LITERALLY					
17			NEGLECTERR	LITERALLY					
17			NESTEDIFERR	LITERALLY					
17			NESTEDINCLUDEERR	LITERALLY					
61	0000H		NEWCMPODY	PROCEDURE BYTE EXTERNAL(17) STACK=0000H					
64	0000H		NEWCMNAME	PROCEDURE BYTE EXTERNAL(18) STACK=0000H					
46	0000H		NEWMACRO	PROCEDURE BYTE EXTERNAL(12) STACK=0000H					
43	0000H		NEWSYMBOL	PROCEDURE BYTE EXTERNAL(11) STACK=0000H					
13			NIL	LITERALLY					

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

119	0000H	1	NOERRPR.	BYTE EXTERNAL(45)	122 124 143 146
34	0000H		NOFORWARDEXPR. . .	PROCEDURE BYTE EXTERNAL(8) STACK=0000H	
29	0000H		NOFORWARDOPER. . .	PROCEDURE BYTE EXTERNAL(4) STACK=0000H	
6			NMOD.	LITERALLY	
119	0000H	1	NOOPER.	BYTE EXTERNAL(48)	179 183 196 200 203
				219	
3			NOOVRCOUNT. . . .	LITERALLY	
3			NOOVRRTDEBIT. . . .	LITERALLY	
8			NOPARAND.	LITERALLY	
11			NORANGE.	LITERALLY	
12			NUMBER.	LITERALLY	223
11			NUMBERRANGE. . . .	LITERALLY	
16			OAND.	LITERALLY	
16			OEQ.	LITERALLY	
177	0006H	2	OFFSET.	WORD MEMBER(OPER)	
119	0006H	2	OFFSET.	WORD MEMBER(OPERANDS)	
205	0006H	2	OFFSET.	WORD MEMBER(OPER)	223
16			OGE.	LITERALLY	
16			OGT.	LITERALLY	
16			OLAST.	LITERALLY	
16			OLE.	LITERALLY	
16			OLENGTH.	LITERALLY	
16			OLT.	LITERALLY	
16			OMOD.	LITERALLY	
16			ONE.	LITERALLY	
16			ONOT.	LITERALLY	
16			OOFFSET.	LITERALLY	
16			ODR.	LITERALLY	
177	0000H	9	OPER.	STRUCTURE BASED(PT)	186
205	0000H	9	OPER.	STRUCTURE BASED(PT)	210 212 221 223
27	0000H		OPERAND.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H	134
119	0000H	36	OPERANDS.	STRUCTURE ARRAY(4) EXTERNAL(49)	183 209
				220	
3			OPERANDSTRUC. . . .	LITERALLY	119 177 205
12			OPERATOR.	LITERALLY	
17			OPMISMATCH.	LITERALLY	244
16			OPTR.	LITERALLY	
16			OSEG.	LITERALLY	
16			OSHL.	LITERALLY	
16			OSHORT.	LITERALLY	
16			OSHR.	LITERALLY	
16			QTYPE.	LITERALLY	
16			QXOR.	LITERALLY	
15			PCODEMACRO.	LITERALLY	
15			PCSEG.	LITERALLY	
15			PDB.	LITERALLY	
15			PDBIT.	LITERALLY	
15			PDD.	LITERALLY	
15			PDSEG.	LITERALLY	
15			PDW.	LITERALLY	
15			PEJECT.	LITERALLY	
15			PEND.	LITERALLY	
15			PENDIF.	LITERALLY	
15			PENDM.	LITERALLY	
15			PEQU.	LITERALLY	
15			PESEG.	LITERALLY	
15			PIF.	LITERALLY	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

[illegible]

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

2		REENT.	LITERALLY		
12		REG.	LITERALLY		
11		REGSTERRANGE.	LITERALLY		
14		RES.	LITERALLY		
58	0000H	2 RESULT	WORD PARAMETER	59	
49	0000H	2 RESULT	WORD PARAMETER	50	
43	0000H	2 RESULT	WORD PARAMETER	44	
64	0000H	2 RETURNPTR.	WORD PARAMETER	65	
16		RIGHTBRACKET	LITERALLY		
14		RSI.	LITERALLY		
14		RSP.	LITERALLY		
14		RSS.	LITERALLY		
24	0000H	2 S.	WORD PARAMETER	25	
69	0000H	SCAN	PROCEDURE EXTERNAL(20) STACK=0000H	165	195
67	0000H	SCANINIT	PROCEDURE EXTERNAL(19) STACK=0000H		
117	0000H	SEARCHFORMATCH	PROCEDURE BYTE EXTERNAL(42) STACK=0000H	122	
3		SEGMBIT.	LITERALLY		
3		SEGMCOUNT.	LITERALLY		
205	0004H	2 SEGMENT.	WORD MEMBER(COPER)		
177	0004H	2 SEGMENT.	WORD MEMBER(COPER)		
119	0004H	2 SEGMENT.	WORD MEMBER(COPERANDS)		
3		SEGTYPEBIT	LITERALLY		
3		SEGTYPECOUNT	LITERALLY		
205	0003H	1 SFLAG.	BYTE MEMBER(COPER)	212	221
177	0003H	1 SFLAG.	BYTE MEMBER(COPER)		
119	0003H	1 SFLAG.	BYTE MEMBER(COPERANDS)		
11		SINGLERANGE.	LITERALLY		
71	0000H	SKIPRESTOFLYNE	PROCEDURE EXTERNAL(21) STACK=0000H	170	257
76	0000H	SKIPUNTIL.	PROCEDURE BYTE EXTERNAL(23) STACK=0000H	189	
55	0000H	2 SOURCE	WORD PARAMETER	56	
2		SPACE.	LITERALLY		
12		SPEC.	LITERALLY		
5		SPECA.	LITERALLY		
5		SPECC.	LITERALLY		
5		SPECD.	LITERALLY		
5		SPECL.	LITERALLY		
73	0000H	SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(22) STACK=0000H	193	
5		SPECM.	LITERALLY		
5		SPECR.	LITERALLY		
5		SPECS.	LITERALLY		
5		SPECX.	LITERALLY		
7		SSOVER	LITERALLY		
58	0000H	2 STRADR	WORD PARAMETER	59	
49	0000H	2 STRADR	WORD PARAMETER	50	
46	0000H	2 STRADR	WORD PARAMETER	47	
43	0000H	2 STRADR	WORD PARAMETER	44	
12		STRING	LITERALLY		
2		STRUC.	LITERALLY	119	177 205
205	0002H	1 STYPE.	BYTE MEMBER(COPER)	210	223
177	0002H	1 STYPE.	BYTE MEMBER(COPER)	186	
119	0002H	1 STYPE.	BYTE MEMBER(COPERANDS)		
55	0000H	2 SYMBADR.	WORD PARAMETER	56	
52	0000H	2 SYMBADR.	WORD PARAMETER	53	
41	0000H	SYMBANIT	PROCEDURE EXTERNAL(10) STACK=0000H		
12		SYMBOL	LITERALLY		
3		SYMBOLHEAD	LITERALLY		
3		SYMBOLSTRUC.	LITERALLY	119	177 205

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

39 0000H      SYMTERMATE. . . . . PROCEDURE EXTERNAL(9) STACK=0000H
2            TAB. . . . . LITERALLY
2            TREND0 . . . . . LITERALLY      122 163 184 193 210 230 232
                                     234 236 238
76 0000H      1 TOK. . . . . BYTE PARAMETER      77
73 0000H      1 TOK. . . . . BYTE PARAMETER      74
2            TRUE . . . . . LITERALLY      174 178 190 197 206 217 222
                                     224
3            TYPEGIT. . . . . LITERALLY      212 221
18 0000H      TYPECALC . . . . . PROCEDURE BYTE EXTERNAL(0) STACK=0000H
3            TYPECTUNT. . . . . LITERALLY
12           UDLSYMB . . . . . LITERALLY      186
17           UDLSYMBOL . . . . . LITERALLY
18 0000H      2 VAL. . . . . WORD PARAMETER      19
12           VARIABLE . . . . . LITERALLY      210
13           WRD. . . . . LITERALLY
21 0000H      WRDTEST. . . . . PROCEDURE BYTE EXTERNAL(1) STACK=0000H      223

```

MODULE INFORMATION:

```

CODE AREA SIZE      = 025FH      607D
CONSTANT AREA SIZE  = 000CH      12D
VARIABLE AREA SIZE  = 000AH      10D
MAXIMUM STACK SIZE  = 0012H      18D
735 LINES READ
0 PROGRAM ERROR(S)

```

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE DECODEL
 OBJECT MODULE PLACED IN OLIVE.OBJ

COMPILER INVOKED BY: :F0: OLIVE.PL4 DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

```
1      %title ("DECODE LINE MODULE")
```

```
      decodel:
```

```
      do;
```

```
      /*
```

```
         modified 3/26/81 R. Silberstein
```

```
         modified 3/30/81 R. Silberstein
```

```
         modified 4/9/81  R. Silberstein
```

```
         modified 4/10/81 R. Silberstein
```

```
         modified 7/24/81 R. Silberstein
```

```
      */
```

```
      /*
```

```
         This is the module to decode each logical sourceline.
         The module takes care of all symbol definitions, and
         activates the PSEUDO-module and the INSTRUCTION-module
         to perform the assembly of the current non-empty source-
         line.
```

```
      */
```

```
      = $include (:f1:macro.lit)
      $nolist
      = $include (:f1:struc.lit)
      $nolist
      = $include (:f1:equals.lit)
      $nolist
      = $include (:f1:ermod.lit)
      $nolist
      = $include (:f1:subr1.ext)
      $nolist
      = $include (:f1:subr2.ext)
      $nolist
      = $include (:f1:scan.ext)
      $nolist
      = $include (:f1:print.ext)
      $nolist
      = $include (:f1:instr.ext)
      $nolist
      = $include (:f1:pseud1.ext)
      $nolist
      = $include (:f1:pseud?.ext)
      $nolist
      = $include (:f1:ermod.ext)
      $nolist
      = $include (:f1:symb.ext)
      $nolist
      = $include (:f1:exglob.ext)
      $nolist
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

    $include (:f1:dline.x86)
= $nolist
    $include (:f1:cm.ext)
= $nolist

```

```

193 1    saveaccum: proc;
194 2        acclensave=acclen;
195 2        call copy(acclen,.accum(0),.accumsave(0));
196 2    end saveaccum;

197 1    exchangeaccum: proc;
198 2        dcl locacclen byte, locaccum(40) byte;
199 2        locacclen=acclensave;
200 2        call copy(acclensave,.accumsave(0),.locaccum(0));
201 2        call saveaccum;
202 2        acclen=locacclen;
203 2        call copy(locacclen,.locaccum(0),.accum(0));
204 2    end exchangeaccum;

205 1    clearsymboll: proc;
206 2        CALL FILL (0, .CURRENTSYMBOL.BASEINDEX-.CURRENTSYMBOL+1, .CURRENTSYMBOL);
207 2    end clearsymboll;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

208 1    pseudotype: proc(lg,ptr) byte;
209 2        dcl (lg,i,lvalue) byte, ptr address, pstable based ptr (1) byte;
210 2        if token.type <> pseudo then return lg+1;
212 2        i=0fff;
213 2        do while (i:=i+1) < lg;
214 3            lvalue=token.value;
215 3            if lvalue=pstable(i) then$do
217 4                call scan; /* skip found pseudo */
218 4                return i;
219 4            end$if;
221 2        end$while;
222 2        return i;
222 2    end pseudotype;

```

/* test if symbol if double defined or "neglected" symbol */

```

223 1    not$doub$negl: proc(errno) byte;
224 2        dcl (errno,errfl) byte;
225 2        if pass = 0 then$do
227 3            if findsymbol(acclensave,.accumsave,.symtabadr) then$do
229 4                call getattributes(symtabadr,.currentsymbol);
230 4                if currentsymbol.stype <> neglected then$do
232 5                    currentsymbol.stype=doubleddefined;
233 5                    call enterattributes(symtabadr,.currentsymbol);
234 5                end$if;
235 4                return false;
236 4            end$if;

```

```

else$do
    /* pass 1 and pass 2 */

239 3      if not findsymbol(acclensave,.accumsave,.symtabadr) then
240 3                                     return false;
241 3      call getattributes(symtabadr,.currentsymbol);
242 3      errfl=true;
243 3      if currentsymbol.stype=neglected then$do
244 4          errno=neglecterr;
245 4      else$do
246 4          if currentsymbol.stype<>doubleddefined then errfl=false;
247 4      end$if;
248 4      if errfl then$do
249 5          call exchangeaccum;
250 5          call errmsg(errno);
251 5          call exchangeaccum;
252 5          return false;
253 5      end$if;
254 2      return true;
255 2      end not$doub$negl;

261 1      newsym: proc byte; /* enter new symbol into table */
262 2          if pass=0 then$do
263 3              if not newsymbol(acclensave,.accumsave,.symtabadr) then$do
264 4                  fullsymtab=true;
265 4                  return false;
266 4              end$if;
267 2          end$if;
268 2          return true;
269 2      end newsym;

/* set up symbol attributes for label,OR,DW,DD,RS */

272 1      setupattr: proc (styp,sfla);
273 2          dcl (styp,sfla,segtyp) byte;
274 2          segtyp=shl(csegtyp,segtypecount) and segtypebit;
275 2          currentsymbol.stype=styp;
276 2          if csegspec then sfla=sfla or segmbit;
277 2          currentsymbol.sflag=sfla or segtyp;
278 2          currentsymbol.segment=csegvalue;
279 2          currentsymbol.offset=cip;
280 2      end setupattr;

282 1      entatr: proc; /* enter attributes of current symbol into table */
283 2          if pass <> 2 then$do
284 3              call enter$attributes(symtabadr,.currentsymbol);
285 3          end$if;
286 2      end entatr;

/* decode instruction */

288 1      decodeinstr: proc;
289 2          if csegtyp <> rcs then$do
290 3              call errmsg(instrerr);
291 3              call skip$rest$of$line;
292 3          else$do
293 3              CALL LISTCIP;
294 3          end$if;

```

CCD/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

```

296 3      call instruction; /* decode instruction */
297 3      endif;
298 2      end decodeinstr;

299 1      labinstruction: proc; /* scan labelled instruction */
300 2          dcl symb based codemacroptr symbolstruc;

301 2          call saveaccum;
                                /* enter label into symbol table */
302 2          if not$doublingl(doubledflab) then$do
304 3              if newsym then$do
306 4                  call setupattr(lab, wrd);
307 4                  call entatr;
308 4              endif;
310 2          call scan;          /* skip ":" */
311 2          call scan;          /* allow empty instruction */
312 2          if emptyline then$do
314 3              call skip$rest$of$line;
315 3          else$do
317 3              if findcodemacro(acclen,.accum(0),.codemacroptr) then$do
319 4                  call scan; /* skip codemacro */
320 4                  call decode$instr;
321 4              else$do
323 4                  call errmsg(illegalmacro);
324 4              endif;
325 3          endif;
326 2      end labinstruction;

327 1      no$ident$pseudo: proc; /* branch to correct pseudo routine */
328 2          dcl ptable(*) byte data( /* define legal unnamed pseudos */
pif,pendif,pinclude,pcseg,pdseg,psseg,peseg,porg,pdb,
pdw,pdd,prb,prs,prw,pend,ppagesize,ppagewidth,
ptitle,peject,psimform,pcodemacro,plist,pnolist,PIFLIST,PNOIFLIST,
psegfix,pnosegfix,pmodrm,prelb,prelw,pdbit,pendm);

329 2          do case pseudotype(length(ptable),.ptable); /* branch */
330 3              call IFrout;
331 3              call ENDFrout;
332 3              call INCLUDErout;
333 3              call CSEGrout;
334 3              call DSEGrout;
335 3              call SSEGrout;
336 3              call ESEGrout;
337 3              call DRGrout;
338 3              if codemacro$flag then call db$cm$rout;
340 3                  else call DBrout;
341 3              if codemacro$flag then call dw$cm$rout;
343 3                  else call DWrout;
344 3              if codemacro$flag then call dd$cm$rout;
346 3                  else call DDRout;
347 3              call RSrout(byt); /* RB */
348 3              call RSrout(byt); /* RS */
349 3              call RSrout(wrd); /* RW */

```

COPYR GHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

350 3      call ENDrout;
351 3      call PAGE$IZrout;
352 3      call PAGE$IDrout;
353 3      call TITLrout;
354 3      call EJECTrout;
355 3      call SIMFORKrout;
356 3      call CODE$MACRrout;
357 3      call LISTrout;
358 3      call NOLISrout;
359 3      CALL IFLISROUT;
360 3      CALL NOTFLISROUT;
361 3      call seqfix$cmrout; /* cm */
362 3      call noseqfix$cmrout; /* cm */
363 3      call modrm$cmrout; /* ca */
364 3      call relb$cmrout; /* cm */
365 3      call relw$cmrout; /* cm */
366 3      call dbit$cmrout; /* cm */
367 3      call end$cmrout; /* cm */
368 3      do; /* error, illegal pseudo */
369 4      call errmsg(illegalpseudo);
370 4      call skip$rest$of$line;
371 4      end;
372 3      end$case;
373 2      end no$ident$pseudo;

374 1      identpseudo: proc(normal); /* scan a named pseudo instruction */
375 2      dcl (noerr,normal) byte,symb based codemacroptr symbolstruc;

376 2      entervar: proc(typ);
377 3      dcl typ byte;
378 3      noerr=false;
379 3      if not$doub$negl(doublededefvar) then$do
381 4      if newsym then$do
383 5      call setupattr(variable,typ);
384 5      noerr=true;
385 5      end$if;
387 3      end$if;
388 2      end enter;
389 3      if noerr then call entatr;
391 3      end enter;

392 2      /* legal pseudos: DB,DW,DD,RB,RS,RW,EQU */
393 2      dcl pseudotable(7) byte data(pdb,pdw,pdd,prb,prs,prw,pequ);

394 2      call clearsymbol; /* clear attributes of current symbol */
395 2      if normal then$do /* unormal if EQU with instruction parameter */
396 3      call saveaccum;
397 3      call scan; /* scan actual pseudo */
398 3      end$if;
399 2      do case pseudotype(length(pseudotable),.pseudotable);
400 3      do; /* DB */
401 4      call entervar(byt);
402 4      call DBrout;
403 4      call enter;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
404 4      end;

405 3      do;                /* DW */
406 4      call entervar(wrd);
407 4      call Dwrout;
408 4      call enter;
409 4      end;

410 3      do;                /* DD */
411 4      call entervar(dwr);
412 4      call DDrou;
413 4      call enter;
414 4      end;

415 3      do;                /* RB */
416 4      call entervar(byt);
417 4      call RSrou;
418 4      call enter;
419 4      end;

420 3      do;                /* RS */
421 4      call entervar(byt);
422 4      call RSrou;
423 4      call enter;
424 4      end;

425 3      do;                /* RW */
426 4      call entervar(wrd);
427 4      call RSrou;
428 4      call enter;
429 4      end;

430 3      do;                /* EQJ */
431 4      if not$doub$neyl(doubled$ymb) then$do
433 5          if newsym then$do
435 6              call EQUrou;
436 6              else$do
438 6                  call skip$rest$of$line;
439 6              end$if;
440 5          else$do
442 5              call skip$rest$of$line;
443 5          end$if;
444 4      end;

445 3      do;                /* illegal pseudo instruction */
446 4      call errmsg(illegalpseudo);
447 4      call skip$rest$of$line;
448 4      end;

449 3      do;                /* missing pseudo instruction */
450 4      call errmsg(missingpseudo);
451 4      call skip$rest$of$line;
452 4      end;
453 3      end$case;
454 2      end identpseudo;

455 1      decodeline: proc public;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____


```

456 2      first$item$type: proc byte;
457 3      decl typ byte;
458 3      typ=token.type;
459 3      if typ=pseudo then return 3;
461 3      if typ=ident and nextch=":" then return 2;
463 3      if (typ=ident) or (typ=operator) then do
465 4          if findcodemacro(acclen,accum(0),codemacro,ptr) then do
467 5              call saveaccum;
468 5              call scan; /* skip found codemacro */
469 5              typ=token.value;
470 5              if (token.type=pseudo) and (typ=pequ) then return 5;
472 5              return 4;
473 5          end$if;
475 3      end$if;
477 3      if typ <> ident then return 0; /* error */
478 3      return 1;
479 3  end first$item$type;

479 2      if accum(0) <> cr then do /* skip blank lines */
481 3          do case first$item$type;
482 4              do; /* error, skip rest of line */
483 5                  call errmsg(first$item); /* error handler */
484 5                  call skip$rest$of$line;
485 5                  end;
486 4                  call ident$pseudo(true); /* named pseudo instruction */
487 4                  call lab$instruction; /* label (followed by instruction) */
488 4                  call no$ident$pseudo; /* pseudo instruction */
489 4                  call decodeinstr; /* code instruction */
490 4                  call ident$pseudo(false); /* EQU with instruction parameter */
491 4              end$case;
492 3          end$if;
493 2      end decode$line;

494 1  end$module decode;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
169	0000H	1	ACCLEN BYTE EXTERNAL(74) 194 195 202 317 465
170	0000H	1	ACCLENSAVE BYTE EXTERNAL(82) 194 199 200 227 233
			264
169	0000H	80	ACCUM. BYTE ARRAY(80) EXTERNAL(75) 195 203 317
			465 479
170	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(81) 195 200 227
			239 264
2		ADDR	LITERALLY 11 14 17 20 26 44 47
			50 53 139 140 146 149 152 155 158 161
			164 167 169 170 300 375
34	0000H		ALPHANUMERIC PROCEDURE BYTE EXTERNAL(8) STACK=0000H
37	0000H		ASCIICHAR. PROCEDURE BYTE EXTERNAL(9) STACK=0000H
166	0000H	2	ASCIIPTR WORD PARAMETER 167
300	0008H	1	BASEINDEX. BYTE MEMBER(SYMB)
170	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL) 206
169	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
375	0008H	1	BASEINDEX. BYTE MEMBER(SYMB)
3			BASEREBIT LITERALLY
3			BASEREGCOUNT LITERALLY
3			BREGBIT. LITERALLY
3			BREGCOUNT. LITERALLY
5			BYT. LITERALLY 347 348 401 416 417 421 422
31	0000H	1	CH BYTE PARAMETER 32
28	0000H	1	CH BYTE PARAMETER 29
34	0000H	1	CH BYTE PARAMETER 35
37	0000H	1	CH BYTE PARAMETER 38
25	0000H	1	CH BYTE PARAMETER 26
69	0000H	1	CH BYTE PARAMETER 70
40	0000H	1	CH BYTE PARAMETER 41
169	0000H	2	CIP. WORD EXTERNAL(66) 280
205	0043H	19	CLEARSYMBOL. PROCEDURE STACK=000AH 393
4			CODE LITERALLY
9			CODEMACROERR. LITERALLY
170	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(87) 338 341 344
3			CODEMACROHEAD. LITERALLY
170	0000H	2	CODEMACROPTR. WORD EXTERNAL(86) 300 317 375 465
171	0000H		CODEMACROUT. PROCEDURE EXTERNAL(88) STACK=0000H 356
16	0000H		COPY PROCEDURE EXTERNAL(2) STACK=0000H 195 200
			203
2			CR LITERALLY 479
98	0000H		CSEGROUT PROCEDURE EXTERNAL(34) STACK=0000H 333
169	0000H	1	CSEGSPEC BYTE EXTERNAL(69) 276
169	0000H	1	CSEGTYP. BYTE EXTERNAL(67) 274 289
169	0000H	2	CSEGV. WORD EXTERNAL(68) 279
169	0000H	2	CURDSEG. WORD EXTERNAL(71)
170	0000H	9	CURRENTSYMBOL. STRUCTURE EXTERNAL(84) 206 229 230 232
			233 241 243 248 275 278 279 280 285
16	0000H	2	D. WORD PARAMETER 17
52	0000H	2	D. WORD PARAMETER 53

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

49	0000H	2	D.	WORD PARAMETER	50		
46	0000H	2	D.	WORD PARAMETER	47		
43	0000H	2	D.	WORD PARAMETER	44		
173	0000H		DBCMROUT	PROCEDURE EXTERNAL(89) STACK=0000H		339	
189	0000H		DBITCMROUT	PROCEDURE EXTERNAL(97) STACK=0000H		366	
89	0000H		DBROUT	PROCEDURE EXTERNAL(30) STACK=0000H		340	402
2			DCL.	LITERALLY			
177	0000H		DDCMROUT	PROCEDURE EXTERNAL(91) STACK=0000H		345	
93	0000H		DDRROUT	PROCEDURE EXTERNAL(32) STACK=0000H		346	412
288	01B2H	29	DECODEINSTR.	PROCEDURE STACK=0006H	320	489	
1	0000H		DECODEL.	PROCEDURE STACK=0060H			
455	0432H	76	DECODELINE	PROCEDURE PUBLIC STACK=0024H			
52	0000H		DECOUT	PROCEDURE EXTERNAL(14) STACK=0000H			
4			DELETEDSYMB.	LITERALLY			
169	0001H	1	DESCR.	BYTE MEMBER(TOKEN)			
154	0000H	2	DEST	WORD PARAMETER	155		
28	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(6) STACK=0000H			
4			DOUBLEDEFINED.	LITERALLY	232	248	
9			DOUBLEDEFLAB	LITERALLY	302		
9			DOUBLEDEFSYMB.	LITERALLY	431		
9			DOUBLEDEFVAR	LITERALLY	379		
100	0000H		DSEGRROUT	PROCEDURE EXTERNAL(35) STACK=0000H		334	
169	0000H	1	DSPEC.	BYTE EXTERNAL(70)			
175	0000H		DWCMROUT	PROCEDURE EXTERNAL(90) STACK=0000H		342	
5			DWRD	LITERALLY	411		
91	0000H		DWRROUT	PROCEDURE EXTERNAL(31) STACK=0000H		343	407
83	0000H		EJECT.	PROCEDURE EXTERNAL(27) STACK=0000H			
124	0000H		EJECTROUT.	PROCEDURE EXTERNAL(47) STACK=0000H		354	
2			ELSEDO	LITERALLY	236		
67	0000H		EMPTYLINE.	PROCEDURE BYTE EXTERNAL(20) STACK=0000H		312	
2			ENDCASE.	LITERALLY			
191	0000H		ENDCMROUT.	PROCEDURE EXTERNAL(98) STACK=0000H		367	
2			ENDDO.	LITERALLY			
2			ENDFOREVER	LITERALLY			
2			ENOIF.	LITERALLY	249	257	268
112	0000H		ENDIFROUT.	PROCEDURE EXTERNAL(41) STACK=0000H	308	385	473
2			ENOMODULE.	LITERALLY		331	
2			ENDOFFILE.	LITERALLY			
9			ENDOFFLINEERR	LITERALLY			
140	0000H	2	ENDOSYMBTAB	WORD EXTERNAL(55)			
2			ENDPROC.	LITERALLY			
116	0000H		ENDROUT.	PROCEDURE EXTERNAL(43) STACK=0000H		350	
2			ENDWHILE	LITERALLY	219		
282	019BH	23	ENTATR	PROCEDURE STACK=0008H	307	390	
388	0423H	15	ENTER.	PROCEDURE STACK=000CH	403	408	413
					423	428	
157	0000H		ENTERATTRIBUTES.	PROCEDURE EXTERNAL(62) STACK=0000H		233	285
376	03FBH	43	ENTERVAR	PROCEDURE STACK=001AH	401	406	411
					421	426	
43	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(11) STACK=0000H			
108	0000H		EQRROUT.	PROCEDURE EXTERNAL(39) STACK=0000H		435	
224	0053H	1	ERRFL.	BYTE	242	249	251
136	0000H		ERRMSG	PROCEDURE EXTERNAL(53) STACK=0000H		254	291
					323	369	446
136	0000H	1	ERRNO.	BYTE PARAMETER	137		
223	0004H	1	ERRNO.	BYTE PARAMETER AUTOMATIC	224	245	254
4			ERROR.	LITERALLY			

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

104	0000H		ESGR0UT	PROCEDURE EXTERNAL(37) STACK=0000H	336
197	0017H	44	EXCHANGEACCU.	PROCEDURE STACK=0000H	253 255
2			FALSE	LITERALLY 235 240 249 256 267	378 490
22	0000H	2	FCBADR	WORD PARAMETER 23	
22	0000H		FILEAROT	PROCEDURE EXTERNAL(4) STACK=0000H	
3			FILEISTRUCTURE	LITERALLY	
3			FILEOS ISTRUCTURE	LITERALLY	
9			FILESYNTERR.	LITERALLY	
25	0000H		FILL	PROCEDURE EXTERNAL(5) STACK=0000H	206
160	0000H		FINDCODEMACRO.	PROCEDURE BYTE EXTERNAL(63) STACK=0000H	317
				465	
151	0000H		FINDSYMBOL	PROCEDURE BYTE EXTERNAL(60) STACK=0000H	227
				239	
9			FIRSTITEM.	LITERALLY 483	
456	047EH	117	FIRSTITEMTYPE.	PROCEDURE BYTE STACK=000EH	481
2			FOREVER.	LITERALLY	
2			FORMFEED	LITERALLY	
139	0000H	2	FREET	WORD EXTERNAL(54)	
170	0000H	1	FULLSYMBTAB.	BYTE EXTERNAL(83) 266	
154	0000H		GETATTRIBUTES.	PROCEDURE EXTERNAL(61) STACK=0000H	229 241
46	0000H		HEX1OUT.	PROCEDURE EXTERNAL(12) STACK=0000H	
49	0000H		HEX2OUT.	PROCEDURE EXTERNAL(13) STACK=0000H	
209	0051H	1	I.	BYTE 212 213 215 218 221	
4			IDENT.	LITERALLY 461 463 475	
374	0340H	171	IDENTPSEUDO.	PROCEDURE STACK=0020H	486 490
132	0000H		IFLISTROUT	PROCEDURE EXTERNAL(51) STACK=0000H	359
9			IFPARERR	LITERALLY	
110	0000H		IFROUT	PROCEDURE EXTERNAL(40) STACK=0000H	330
9			ILLEGALMACRO	LITERALLY 323	
9			ILLEGALPSEUDO.	LITERALLY 369 446	
9			ILLEXPRELEM.	LITERALLY	
9			ILLIOPER	LITERALLY	
114	0000H		INCLUDEROUT.	PROCEDURE EXTERNAL(42) STACK=0000H	332
3			INDEXREGBIT.	LITERALLY	
3			INDEXREGCOUNT.	LITERALLY	
2			INIT	LITERALLY	
9			INSTRERR	LITERALLY 291	
85	0000H		INSTRUCTION.	PROCEDURE EXTERNAL(28) STACK=0000H	296
3			IREGBIT.	LITERALLY	
3			IREGCOUNT.	LITERALLY	
4			LAB.	LITERALLY 306	
299	01CFH	88	LABINSTRUCTION	PROCEDURE STACK=0018H	487
9			LABOUTOFRANGE.	LITERALLY	
8			LEFTBRACKET.	LITERALLY	
			LENGTH	BUILTIN 329 399	
169	0000H	2	LENGTH	WORD MEMBER(OPERANDS)	
300	0000H	2	LENGTH	WORD MEMBER(SYMB)	
170	0000H	2	LENGTH	WORD MEMBER(CURRENTSYMBOL)	
375	0000H	2	LENGTH	WORD MEMBER(SYMB)	
31	0000H		LETTER	PROCEDURE BYTE EXTERNAL(7) STACK=0000H	
2			LF	LITERALLY	
208	0006H	1	LG	BYTE PARAMETER AUTOMATIC	209 211 213
160	0000H	1	LG	BYTE PARAMETER 161	
151	0000H	1	LG	BYTE PARAMETER 152	
148	0000H	1	LG	BYTE PARAMETER 149	
163	0000H	1	LG	BYTE PARAMETER 164	
145	0000H	1	LG	BYTE PARAMETER 146	

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

166	0000H	1	LG	BYTE PARAMETER	167					
87	0000H		LSTCIP.	PROCEDURE EXTERNAL(29) STACK=0000H		295				
128	0000H		LSTROUT	PROCEDURE EXTERNAL(49) STACK=0000H		357				
2			LIT.	LITERALLY	2	3	4	5	6	7
					9					
198	0000H	1	LOCACLEN.	BYTE	199	202	203			
198	0001H	80	LOCACCOM	BYTE ARRAY(80)		200	203			
209	0052H	1	LVALUE	BYTE	214	215				
148	0000H	2	MACDEFT	WORD PARAMETER		149				
9			MISSIFERR.	LITERALLY						
9			MISSINGPSEUDO.	LITERALLY	450					
9			MISSINSTR.	LITERALLY						
9			MISSSGMINFD	LITERALLY						
9			MISSYPEANFD	LITERALLY						
183	0000H		MODRMCHROUT.	PROCEDURE EXTERNAL(94) STACK=0000H		363				
25	0000H	1	N.	BYTE PARAMETER	26					
16	0000H	1	N.	BYTE PARAMETER	17					
13	0000H	2	N.	WORD PARAMETER	14					
52	0000H	2	N.	WORD PARAMETER	53					
49	0000H	2	N.	WORD PARAMETER	50					
46	0000H	1	N.	BYTE PARAMETER	47					
43	0000H	1	N.	BYTE PARAMETER	44					
4			NEGLECTED.	LITERALLY	230	243				
9			NEGLECTERR.	LITERALLY	245					
9			NESTEDIFERR.	LITERALLY						
9			NESTEDINCLUDEERR.	LITERALLY						
163	0000H		NEWCMRDY.	PROCEDURE BYTE EXTERNAL(64) STACK=0000H						
166	0000H		NEWCMNAME.	PROCEDURE BYTE EXTERNAL(65) STACK=0000H						
148	0000H		NEWMACRO	PROCEDURE BYTE EXTERNAL(59) STACK=0000H						
261	0136H	44	NEWSYM	PROCEDURE BYTE STACK=000AH	304					
145	0000H		NEWSYMBOL.	PROCEDURE BYTE EXTERNAL(58) STACK=0000H		381	433			
169	0000H	1	NEXTCH	BYTE EXTERNAL(73)	461					
5			NIL.	LITERALLY						
375	0055H	1	NOERR.	BYTE	378	384	389			
327	0227H	294	NOIDENTPSEUDO.	PROCEDURE STACK=000CH	488					
134	0000H		NOIFLISTROUT	PROCEDURE EXTERNAL(52) STACK=0000H		360				
130	0000H		NOLISTROUT	PROCEDURE EXTERNAL(50) STACK=0000H		358				
169	0000H	1	NOOPER	BYTE EXTERNAL(76)						
3			NOOVERCOUNT.	LITERALLY						
3			NOOVERRIDEBIT.	LITERALLY						
374	0004H	1	NORMAL	BYTE PARAMETER AUTOMATIC	375	394				
181	0000H		NOSEGFIXCKROUT	PROCEDURE EXTERNAL(93) STACK=0000H		362				
223	009AH	156	NOTDOUGNEGL.	PROCEDURE BYTE STACK=0014H	302	379	431			
4			NUMBER	LITERALLY						
8			OAND	LITERALLY						
8			OEQ.	LITERALLY						
375	0006H	2	OFFSET	WORD MEMBER(SYMB)						
170	0006H	2	OFFSET	WORD MEMBER(CURRENTSYMBOL)	280					
169	0006H	2	OFFSET	WORD MEMBER(OPERANDS)						
300	0006H	2	OFFSET	WORD MEMBER(SYMB)						
8			OGE.	LITERALLY						
8			OGT.	LITERALLY						
8			OLAST.	LITERALLY						
8			OLE.	LITERALLY						
8			OLENGTH.	LITERALLY						
8			OLT.	LITERALLY						
8			OMOD	LITERALLY						

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

8		ONE	LITERALLY	
8		ONDT	LITERALLY	
8		OFFSET	LITERALLY	
8		OPR	LITERALLY	
169	0000H	36 OPERANDS	STRUCTURE ARRAY(4) EXTERNAL(77)	
3		OPERANDSTRUC	LITERALLY	169
4		OPERATOR	LITERALLY	463
9		OPMISMATCH	LITERALLY	
8		OPTR	LITERALLY	
106	0000H	ORGRDOUT	PROCEDURE EXTERNAL(38) STACK=0000H	337
8		OSEG	LITERALLY	
8		OSHL	LITERALLY	
8		OSHORT	LITERALLY	
8		OSHR	LITERALLY	
8		OTYPE	LITERALLY	
19	0000H	OUTTEXT	PROCEDURE EXTERNAL(3) STACK=0000H	
8		UXOR	LITERALLY	
118	0000H	PAGESIZEROUT	PROCEDURE EXTERNAL(44) STACK=0000H	351
120	0000H	PAGEWIDTHROUT	PROCEDURE EXTERNAL(45) STACK=0000H	352
170	0000H	1 PASS	BYTE EXTERNAL(78) 225 262 283	
7		PCODEMACRO	LITERALLY	328
7		PCSEG	LITERALLY	328
7		PDB	LITERALLY	328 392
7		PDBIT	LITERALLY	326
7		PDD	LITERALLY	328 392
7		PDSEG	LITERALLY	328
7		PDW	LITERALLY	328 392
7		PEJECT	LITERALLY	328
7		PEND	LITERALLY	328
7		PENDIF	LITERALLY	328
7		PENDM	LITERALLY	328
7		PEQU	LITERALLY	392 470
7		PESEG	LITERALLY	328
7		PIF	LITERALLY	328
7		PIFLIST	LITERALLY	328
7		PTINCLUDE	LITERALLY	328
7		PLIST	LITERALLY	328
7		PMDRM	LITERALLY	328
7		PNOIFLIST	LITERALLY	328
7		PNOLIST	LITERALLY	328
7		PNOSEGFIX	LITERALLY	328
7		PORG	LITERALLY	328
7		PPAGESIZE	LITERALLY	328
7		PPAGEWIDTH	LITERALLY	328
7		PRB	LITERALLY	328 392
170	0000H	240 PREFIX	BYTE ARRAY(240) EXTERNAL(79)	
170	0000H	1 PREFIXPTR	BYTE EXTERNAL(80)	
7		PRELB	LITERALLY	328
7		PRELW	LITERALLY	328
72	0000H	PRINTCRLF	PROCEDURE EXTERNAL(22) STACK=0000H	
78	0000H	PRINTINIT	PROCEDURE EXTERNAL(25) STACK=0000H	
74	0000H	PRINTNEWPAGE	PROCEDURE EXTERNAL(23) STACK=0000H	
69	0000H	PRINTSINGLEBYTE	PROCEDURE EXTERNAL(21) STACK=0000H	
76	0000H	PRINTSOURCELINE	PROCEDURE EXTERNAL(24) STACK=0000H	
80	0000H	PRINTTERMINATE	PROCEDURE EXTERNAL(26) STACK=0000H	
2		PROC	LITERALLY	10 13 34 37 46 49 52
				55 57 59 61 64 67 69 72 74 76

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

			78	80	83	85	89	91	93	95	98	100
			102	104	106	108	110	112	114	116	118	120
			122	124	126	128	130	132	134	136	141	143
			145	148	151	154	157	160	163	166	171	173
			175	177	179	181	183	185	187	189	191	193
			197	205	208	223	261	272	282	288	299	327
			374	376	388	455	456					
7		PRS.	LITERALLY			328	392					
7		PRW.	LITERALLY			328	392					
7		PSEGFIX.	LITERALLY			328						
4		PSEUDO	LITERALLY			210	459	470				
9		PSEUDODUPR.	LITERALLY									
392	0020H	7 PSEUDOTABLE.	BYTE ARRAY(7) DATA				399					
208	0056H	68 PSEUDOTYPE	PROCEDURE BYTE STACK=0008H					329	399			
7		PSIMFORM	LITERALLY			328						
7		PSSEG.	LITERALLY			328						
209	0000H	1 PSTABLE.	BYTE BASED(PTR) ARRAY(1)					215				
25	0000H	2 PT	WORD PARAMETER			26						
328	0000H	32 PTABLE	BYTE ARRAY(32) DATA				329					
7		PTITLE	LITERALLY			328						
208	0004H	2 PTR.	WORD PARAMETER AUTOMATIC					209	215			
163	0000H	2 PTR.	WORD PARAMETER			164						
6		RAH.	LITERALLY									
6		RAL.	LITERALLY									
6		RAX.	LITERALLY									
6		RBH.	LITERALLY									
6		RBL.	LITERALLY									
6		RBP.	LITERALLY									
6		RBX.	LITERALLY									
6		RCH.	LITERALLY									
6		RCL.	LITERALLY									
6		RCS.	LITERALLY									
6		RCX.	LITERALLY									
6		RDH.	LITERALLY									
6		ROL.	LITERALLY									
6		RDL.	LITERALLY									
6		RDS.	LITERALLY									
6		RDX.	LITERALLY									
2		REENT.	LITERALLY									
4		REG.	LITERALLY									
185	0000H	RELBCMRNUT	PROCEDURE EXTERNAL(95) STACK=0000H					364				
187	0000H	RELWCMRUT	PROCEDURE EXTERNAL(96) STACK=0000H					365				
6		RES.	LITERALLY									
160	0000H	2 RESULT	WORD PARAMETER			161						
151	0000H	2 RESULT	WORD PARAMETER			152						
145	0000H	2 RESULT	WORD PARAMETER			146						
166	0000H	2 RETURNPTR.	WORD PARAMETER			167						
8		RIGHTBRACKET	LITERALLY									
6		RSI.	LITERALLY									
6		RSP.	LITERALLY									
95	0000H	RSROJT	PROCEDURE EXTERNAL(33) STACK=0000H					347	348			
			349 417 422 427									
6		RSS.	LITERALLY									
16	0000H	2 S.	WORD PARAMETER			17						
43	0000H	2 S.	WORD PARAMETER			44						
193	0000H	23 SAVEACCU.	PROCEDURE STACK=0004H				201	301	396	467		
57	0000H	SCAN	PROCEDURE EXTERNAL(16) STACK=0000H					217	310			

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

			311 319 397 468		
55	0000H	SCALINIT	PROCEDURE EXTERNAL(15) STACK=0000H		
179	0000H	SEGEXCROUT	PROCEDURE EXTERNAL(92) STACK=0000H	361	
3		SEGMIT	LITERALLY 277		
3		SEGMCOUNT	LITERALLY		
375	0004H	2 SEGMENT	WORD MEMBER(SYMB)		
300	0004H	2 SEGMENT	WORD MEMBER(SYMB)		
170	0004H	2 SEGMENT	WORD MEMBER(CURRENTSYMBOL)	279	
169	0004H	2 SEGMENT	WORD MEMBER(OPERANDS)		
273	0054H	1 SEGTYP	BYTE 274 278		
3		SEGTYPEBIT	LITERALLY 274		
3		SEGTYPECOUNT	LITERALLY 274		
272	0162H	57 SETUPATR	PROCEDURE STACK=0006H	306 383	
272	0004H	1 SFLA	BYTE PARAMETER AUTOMATIC	273 277 278	
375	0003H	1 SFLAG	BYTE MEMBER(SYMB)		
300	0003H	1 SFLAG	BYTE MEMBER(SYMB)		
170	0003H	1 SFLAG	BYTE MEMBER(CURRENTSYMBOL)	278	
169	0003H	1 SFLAG	BYTE MEMBER(OPERANDS)		
		SHL	BUILTIN 274		
126	0000H	SIMFORMROUT	PROCEDURE EXTERNAL(48) STACK=0000H	355	
59	0000H	SKIPRESTUFLINE	PROCEDURE EXTERNAL(17) STACK=0000H	292 314	
			370 438 442 447 451 484		
64	0000H	SKIPUNITL	PROCEDURE BYTE EXTERNAL(19) STACK=0000H		
157	0000H	2 SOURCE	WORD PARAMETER 158		
2		SPACE	LITERALLY		
4		SPEC	LITERALLY		
61	0000H	SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(18) STACK=0000H		
102	0000H	SSEGROUT	PROCEDURE EXTERNAL(36) STACK=0000H	335	
160	0000H	2 STRADR	WORD PARAMETER 161		
151	0000H	2 STRADR	WORD PARAMETER 152		
148	0000H	2 STRADR	WORD PARAMETER 149		
145	0000H	2 STRADR	WORD PARAMETER 146		
4		STRING	LITERALLY		
2		STRUC	LITERALLY 169 170 300 375		
272	0006H	1 STYP	BYTE PARAMETER AUTOMATIC	273 275	
375	0002H	1 STYPE	BYTE MEMBER(SYMB)		
300	0002H	1 STYPE	BYTE MEMBER(SYMB)		
170	0002H	1 STYPE	BYTE MEMBER(CURRENTSYMBOL)	230 232 243	
			248 275		
169	0002H	1 STYPE	BYTE MEMBER(OPERANDS)		
375	0000H	9 SYMB	STRUCTURE BASED(CODEMACROPTR)		
300	0000H	9 SYMB	STRUCTURE BASED(CODEMACROPTR)		
157	0000H	2 SYMBADR	WORD PARAMETER 158		
154	0000H	2 SYMBADR	WORD PARAMETER 155		
143	0000H	SYMBINIT	PROCEDURE EXTERNAL(57) STACK=0000H		
4		SYMBOL	LITERALLY		
3		SYMBOLHEAD	LITERALLY		
3		SYMBOLSTRUC	LITERALLY 169 170 300 375		
170	0000H	2 SYMBTABADR	WORD EXTERNAL(85)	227 229 233 239 241	
			264 285		
141	0000H	SYMBTERMINATE	PROCEDURE EXTERNAL(56) STACK=0000H		
19	0000H	2 T	WORD PARAMETER 20		
2		TAB	LITERALLY		
22	0000H	2 TEXTADR	WORD PARAMETER 23		
2		THEND	LITERALLY 215 225 227 230 243 251 262		
			264 283 289 302 304 312 317 319 381 394		
			431 433 463 465 479		

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

122	0000H		TITLEROUT.	PROCEDURE EXTERNAL(46) STACK=0000H	353
64	0000H	1	TOK.	BYTE PARAMETER	65
61	0000H	1	TOK.	BYTE PARAMETER	67
169	0000H	4	TOKEN.	STRUCTURE EXTERNAL(72)	210 214 458 463
				470	
2			TRUE	LITERALLY	242 259 266 270 384 436
457	0056H	1	TYP.	BYTE	458 459 461 463 469 470 475
376	0004H	1	TYP.	BYTE PARAMETER AUTOMATIC	377 393
95	0000H	1	TYP.	BYTE PARAMETER	96
169	0000H	1	TYPE	BYTE MEMBER(TOKEN)	210 458 470
3			TYPEBIT.	LITERALLY	
10	0000H		TYPECALC.	PROCEDURE BYTE EXTERNAL(0) STACK=0000H	
3			TYPECOUNT.	LITERALLY	
4			UDEFSYMB.	LITERALLY	
9			UDEFSYMBOL.	LITERALLY	
40	0000H		UPPER.	PROCEDURE BYTE EXTERNAL(10) STACK=0000H	
80	0000H	1	USEFACT.	BYTE PARAMETER	81
10	0000H	2	VAL.	WORD PARAMETER	11
169	0002H	2	VALUE.	WORD MEMBER(TOKEN)	214 469
4			VARIABLE.	LITERALLY	383
5			WRD.	LITERALLY	306 349 406 426 427
13	0000H		WRDTEST.	PROCEDURE BYTE EXTERNAL(1) STACK=0000H	

MODULE INFORMATION:

CODE AREA SIZE = 04F3H 1267D
 CONSTANT AREA SIZE = 0027H 39D
 VARIABLE AREA SIZE = 0057H 87D
 MAXIMUM STACK SIZE = 0024H 36D
 1005 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____



ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE GLOBAL
OBJECT MODULE PLACED IN GLOBAL.OBJ

COMPILER INVOKED BY: :F0: GLOBAL.PLM DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

```
1      %title ("GLOBAL VARIABLES")
```

```
      global:
```

```
      do;
```

```
      /*
```

```
         modified 3/28/81 R. Silberstein
```

```
         modified 4/16/81 R. Silberstein
```

```
         modified 4/20/81 R. Silberstein
```

```
         modified 7/24/81 R. Silberstein
```

```
         modified 9/2/81 R. Silberstein
```

```
      */
```

```
      /*
```

```
         This module defines all the global variables  
         of the assembler.
```

```
      */
```

```
      %include (:f1:macro.lit)
```

```
=      %nolist
```

```
      %include (:f1:struc.lit)
```

```
=      %nolist
```

```
      %INCLUDE (:F1:SUBP2.EXT)
```

```
=      %nolist
```

```
40 1      dcl
```

```
         /* dummy structure forces contiguous storage */
```

```
      glob      structure (
```

```
      pass      byte, /* current pass no, 1,2,3 */
```

```
         /* address counters */
```

```
      cip      addr, /* current instruction pointer */
```

```
      csegtype byte, /* current segment type, code, data,  
                     stack or extra data */
```

```
      csegvalue addr, /* current segment value */
```

```
      csegspec byte, /* true if segment value specified */
```

```
      escip     addr, /* current ES instruction pointer */
```

```
      cscip     addr, /* current CS instruction pointer */
```

```
      sscip     addr, /* current SS instruction pointer */
```

```
      dscip     addr, /* current DS instruction pointer */
```

```
      curcseg   addr, /* current code segment value */
```

```
      curdseg   addr, /* current data segment value */
```

```
      cursseg   addr, /* current stack segment value */
```

```
      cureseg   addr, /* current extra segment value */
```

```
      cspec     byte, /* true if code segment value given */
```

```
      dspec     byte, /* true if data segment value given */
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

sspec      byte, /* true if stack segment value given */
espec      byte, /* true if extra segment value given */

```

```
/* print output parameters */
```

```

print$on   byte, /* on/off flag */
print$switchoff byte, /* set/reset by NOLIST/LIST */
IFLIST     BYTE, /* SET/RESET BY IFLIST/NOIFLIST */
maxcol     byte); /* pagewidth */

```

```
41 1 dcl
```

```

sourcename (12) byte public, /* source file name */
sourc$stop byte, /* used to contain zero */
savesource (12) byte public, /* source file during INCLUDE file */
print$device byte public, /* print file device */
SYMBOLDEVICE BYTE PUBLIC, /* SYMBOL FILE DEVICE */
title (30) byte public, /* user specified program title */
stop$title byte, /* used to contain zero */
pagesize byte public, /* page size */
simform byte public, /* true if formfeed is to be simulated */
sourcebuf (80) byte public, /* copy of source input to be printed */
sourceptr byte public, /* source buffer pointer */
prefix (240) byte public, /* prefix to source line */
prefixptr byte public, /* pointer to prefix buffer */
ABSADDR (4) BYTE PUBLIC; /* ABSOLUTE ADDRESS FIELD */

```

```
/* references to glob structure */
```

```

42 1 dcl
pass      byte public at(.glob.pass),
cip       addr public at(.glob.cip),
csegtype  byte public at(.glob.csegtype),
csegvalue addr public at(.glob.csegvalue),
cseg$spec byte public at(.glob.cseg$spec),
escip     addr public at(.glob.escip),
cscip     addr public at(.glob.cscip),
sscip     addr public at(.glob.sscip),
dscip     addr public at(.glob.dscip),
curcseg   addr public at(.glob.curcseg),
curdseg   addr public at(.glob.curdseg),
cursseg   addr public at(.glob.cursseg),
cureseg   addr public at(.glob.cureseg),
cspec     byte public at(.glob.cspec),
dspec     byte public at(.glob.dspec),
sspec     byte public at(.glob.sspec),
espec     byte public at(.glob.espec),
print$on  byte public at(.glob.print$on),
print$switchoff byte public at(.glob.print$switchoff),
IFLIST    BYTE PUBLIC AT (.GLOB.IFLIST),
maxcol    byte public at(.glob.maxcol);

```

```
/* io error stpublic atus */
```

```
43 1 dcl
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

errors      addr public,      /* counts no of errors */

/* scanner variables: */

token struct /* actual token scanned */
type byte,   /* token type, legal values :
              reg      - register
              pseudo   - pseudo code
              string   - text string
              spec     - special character
              number   - number
              operator - arithmetic operator
              ident    - identifier */

descr byte, /* token description, legal values :
            nil      - no specification
            byte     - 8 bit type
            word     - 16 bit type
            dword    - 32 bit type */

value addr public, /* token value */
nextch byte public, /* next input character (lookahead) */
acclen byte public, /* accumulator length */
accum(80) byte public, /* actual token scanned */
accmsave(80) byte public, /* used to save accumulator */
acclensave byte public,
eofset byte public, /* true if end-of-file found */

```

```
/* Miscellaneous variables: */
```

```

intel$hex$on byte public, /* true if INTEL hex output format */
noerror      byte public, /* errorflag in codemacro decoding */
errorprinted byte public, /* true if an error is printed */
firstmacroptr address public, /* pointer at first codemacro */
macroptr     address public, /* current pointer within macros */
fullsymtab   byte public, /* true if symboltable is full */
include$on   byte public, /* true if input from INCLUDE file */
IFLEVEL      BYTE PUBLIC, /* IF-ENDIF NESTING LEVEL */
currentsymbol symbolstruc /* current scanned symbol */
              public,
sybtabadr    address public, /* pointer at symbol in table */
nooper       byte public, /* no of instruction operands */
operands(4)  operandstruc /* instruction operands,max 4 */
              public,
codemacroptr address public, /* pointer to found codemacro */
help(5)      byte public, /* scratch area for ascii numbers */
helpstop     byte,
i            byte public, /* scratch variable */
default$drive byte public, /* default disk drive */
include$default byte public, /* default disk for include files */
codemacro$flag byte public; /* true if building a codemacro */

```

```

44 1 globalinit: procedure public; /* initiate some global variables */
45 2   stopitle,sourcstop,helpstop=0;
46 2   pagesize=66;
47 2   fullsymtab,intel$hex$on=false;
48 2   CALL FILL (0, SIZE (TITLE), .TITLE);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
49 2      codemacro$flag=false;  
50 2      end globalinit;
```

```
51 1      end$module global;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

41	01A0H	4	ABSADDR.	BYTE ARRAY(4) PUBLIC	
43	01B0H	1	ACCLN.	BYTE PUBLIC	
43	0251H	1	ACCLNSAVE.	BYTE PUBLIC	
43	01B1H	80	ACCUM.	BYTE ARRAY(80) PUBLIC	
43	0201H	80	ACCUMSAVE.	BYTE ARRAY(80) PUBLIC	
2			ADDR.	LITERALLY	5 11 29 32 35 38 40
				42 43	
19	0000H		ALPHANUMERIC.	PROCEDURE BYTE EXTERNAL(5) STACK=0000H	
22	0000H		ASCIICHAR.	PROCEDURE BYTE EXTERNAL(6) STACK=0000H	
43	0008H	1	BASEINDEX.	BYTE MEMBER(CURRENTSYMBOL)	
43	0008H	1	BASEINDEX.	BYTE MEMBER(COPLRANDS)	
3			BASEREGBIT.	LITERALLY	
3			BASEREGCOUNT.	LITERALLY	
3			BREGBIT.	LITERALLY	
3			BREGCOUNT.	LITERALLY	
16	0000H	1	CH.	BYTE PARAMETER	17
13	0000H	1	CH.	BYTE PARAMETER	14
25	0000H	1	CH.	BYTE PARAMETER	26
10	0000H	1	CH.	BYTE PARAMETER	11
19	0000H	1	CH.	BYTE PARAMETER	20
22	0000H	1	CH.	BYTE PARAMETER	23
40	0001H	2	CIP.	WORD MEMBER(GLOB)	42
42	000FH	2	CIP.	WORD PUBLIC AT	
43	0290H	1	CODEMACROFLAG.	BYTE PUBLIC	49
3			CODEMACROHEAD.	LITERALLY	
43	000CH	2	CODEMACROPTR.	WORD PUBLIC	
2			CR.	LITERALLY	
42	0017H	2	CSCIP.	WORD PUBLIC AT	
40	0009H	2	CSCIP.	WORD MEMBER(GLOB)	42
42	0014H	1	CSEGSPEC.	BYTE PUBLIC AT	
40	0006H	1	CSEGSPEC.	BYTE MEMBER(GLOB)	42
42	0011H	1	CSEGTYP.	BYTE PUBLIC AT	
40	0003H	1	CSEGTYP.	BYTE MEMBER(GLOB)	42
42	0012H	2	CSEGV.	WORD PUBLIC AT	
40	0004H	2	CSEGV.	WORD MEMBER(GLOB)	42
42	0025H	1	CSPEC.	BYTE PUBLIC AT	
40	0017H	1	CSPEC.	BYTE MEMBER(GLOB)	42
42	0010H	2	CURCSEG.	WORD PUBLIC AT	
40	000FH	2	CURCSEG.	WORD MEMBER(GLOB)	42
42	001FH	2	CURDSEG.	WORD PUBLIC AT	
40	0011H	2	CURDSEG.	WORD MEMBER(GLOB)	42
42	0023H	2	CURESEG.	WORD PUBLIC AT	
40	0015H	2	CURESEG.	WORD MEMBER(GLOB)	42
43	0259H	9	CURRENTSYMBOL.	STRUCTURE PUBLIC	
40	0013H	2	CURSSEG.	WORD MEMBER(GLOB)	42
42	0021H	2	CURSSEG.	WORD PUBLIC AT	
28	0000H	2	D.	WORD PARAMETER	29
37	0000H	2	D.	WORD PARAMETER	38
34	0000H	2	D.	WORD PARAMETER	35

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

31	0000H	2	D.	WORD PARAMETER	32
2			DCL.	LITERALLY	
37	0000H		DECOUT	PROCEDURE EXTERNAL(11) STACK=0000H	
43	028EH	1	DEFAULTDRIVE	BYTE PUBLIC	
43	0001H	1	DESCR.	BYTE MEMBER(TOKEN)	
13	0000H		DIGIT.	PROCEDURE BYTE EXTERNAL(3) STACK=0000H	
40	0000H	2	DSCIP.	WORD MEMBER(GLOB)	42
42	0018H	2	DSCIP.	WORD PUBLIC AT	
40	0018H	1	DSPEC.	BYTE MEMBER(GLOB)	42
42	0026H	1	DSPEC.	BYTE PUBLIC AT	
2			ELSEDD	LITERALLY	
2			ENDCASE.	LITERALLY	
2			ENDDO.	LITERALLY	
2			ENDFOREVER	LITERALLY	
2			ENDIF.	LITERALLY	
2			ENDMODULE.	LITERALLY	
2			ENDOFFILE.	LITERALLY	
2			ENDPROC.	LITERALLY	
2			ENDWHTLE	LITERALLY	
43	0252H	1	EOFSET	BYTE PUBLIC	
28	0000H		EQUAL.	PROCEDURE BYTE EXTERNAL(8) STACK=0000H	
43	0255H	1	ERRORPRINTED	BYTE PUBLIC	
43	0000H	2	ERRORS	WORD PUBLIC	
42	0015H	2	ESCIPT.	WORD PUBLIC AT	
40	0007H	2	ESCTP.	WORD MEMBER(GLOB)	42
42	0028H	1	ESPEC.	BYTE PUBLIC AT	
40	001AH	1	ESPEC.	BYTE MEMBER(GLOB)	42
2			FALSE.	LITERALLY	47 49
7	0000H	2	FCBADR	WORD PARAMETER	8
7	0000H		FILEABORT.	PROCEDURE EXTERNAL(1) STACK=0000H	
3			FILEISTRUCTURE	LITERALLY	
3			FILEOSTRUCTURE	LITERALLY	
10	0000H		FILL	PROCEDURE EXTERNAL(2) STACK=0000H	48
43	0006H	2	FIRSTMACROPTR.	WORD PUBLIC	
2			FOREVER.	LITERALLY	
2			FORMFEED	LITERALLY	
43	0256H	1	FULLSYMBTAB.	BYTE PUBLIC	47
40	000EH	31	GLOB	STRUCTURE	42
1	0000H		GLOBAL	PROCEDURE STACK=0000H	
44	0000H	44	GLOBALINIT	PROCEDURE PUBLIC STACK=0000H	
43	0287H	5	HELP	BYTE ARRAY(5) PUBLIC	
43	028CH	1	HELPSTOP	BYTE	45
31	0000H		HEX1OUT.	PROCEDURE EXTERNAL(9) STACK=0000H	
34	0000H		HEX2OUT.	PROCEDURE EXTERNAL(10) STACK=0000H	
43	028DH	1	I.	BYTE PUBLIC	
43	0258H	1	IFLEVEL.	BYTE PUBLIC	
42	002BH	1	IFLIST	BYTE PUBLIC AT	
40	001DH	1	IFLIST	BYTE MEMBER(GLOB)	42
43	028FH	1	INCLUDEDEFAULT	BYTE PUBLIC	
43	0257H	1	INCLUDEON.	BYTE PUBLIC	
3			INDEXREGBIT.	LITERALLY	
3			INDEXREGCOUNT.	LITERALLY	
2			INIT	LITERALLY	
43	0253H	1	INTELHEXON	BYTE PUBLIC	47
3			IREGBIT.	LITERALLY	
3			IREGCOUNT.	LITERALLY	
43	0000H	2	LENGTH	WORD MEMBER(OPERANDS)	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

43	0000H	2	LENGTH	WORD MEMBER(CURRENTSYMBOL)	
16	0000H		LETTER	PROCEDURE BYTE EXTERNAL(4) STACK=0000H	
2			LF	LITERALLY	
2			LIT.	LITERALLY	2 3
43	0008H	2	MACROPTR	WORD PUBLIC	
42	002CH	1	MAXCOL	BYTE PUBLIC AT	
40	001EH	1	MAXCOL	BYTE MEMBER(GLOB)	42
10	0000H	1	N.	BYTE PARAMETER	11
37	0000H	2	N.	WORD PARAMETER	38
34	0000H	2	N.	WORD PARAMETER	35
31	0000H	1	N.	BYTE PARAMETER	32
28	0000H	1	N.	BYTE PARAMETER	29
43	01AFH	1	NEXTCH	BYTE PUBLIC	
43	0254H	1	NOLRROR.	BYTE PUBLIC	
43	0262H	1	NOOPER	BYTE PUBLIC	
3			NOOVERCOUNT.	LITERALLY	
3			NOOVERRIDEBIT.	LITERALLY	
43	0006H	2	OFFSET	WORD MEMBER(CURRENTSYMBOL)	
43	0006H	2	OFFSET	WORD MEMBER(OPERANDS)	
43	0263H	36	OPERANDS	STRUCTURE ARRAY(4) PUBLIC	
3			OPERANDSTRUC	LITERALLY	43
4	0000H		OUTTEXT.	PROCEDURE EXTERNAL(0) STACK=0000H	
41	0067H	1	PAGESIZE	BYTE PUBLIC	46
42	000EH	1	PASS	BYTE PUBLIC AT	
40	0000H	1	PASS	BYTE MEMBER(GLOB)	42
41	00BAH	240	PREFIX	BYTE ARRAY(240) PUBLIC	
41	01AAH	1	PREFIXPTR.	BYTE PUBLIC	
41	0046H	1	PRINTDEVICE.	BYTE PUBLIC	
42	0029H	1	PRINTON.	BYTE PUBLIC AT	
40	001BH	1	PRINTON.	BYTE MEMBER(GLOB)	42
42	002AH	1	PRINTSWITCHOFF.	BYTE PUBLIC AT	
40	001CH	1	PRINTSWITCHOFF.	BYTE MEMBER(GLOB)	42
2			PROC	LITERALLY	19 22 31 34 37
10	0000H	2	PT	WORD PARAMETER	11
2			REENT.	LITERALLY	
28	0000H	2	S.	WORD PARAMETER	29
41	003AH	12	SAVESOURCE	BYTE ARRAY(12) PUBLIC	
3			SEGMBIT.	LITERALLY	
3			SEGMCOUNT.	LITERALLY	
43	0004H	2	SEGMENT.	WORD MEMBER(OPERANDS)	
43	0004H	2	SEGMENT.	WORD MEMBER(CURRENTSYMBOL)	
3			SEGTYPEBIT	LITERALLY	
3			SEGTYPECOUNT.	LITERALLY	
43	0003H	1	SFLAG.	BYTE MEMBER(OPERANDS)	
43	0003H	1	SFLAG.	BYTE MEMBER(CURRENTSYMBOL)	
41	0068H	1	SIHFORM.	BYTE PUBLIC	
			SIZE	BUILTIN	48
41	0069H	80	SOURCEBUF.	BYTE ARRAY(80) PUBLIC	
41	002DH	12	SOURCENAME	BYTE ARRAY(12) PUBLIC	
41	00B9H	1	SOURCEPTR.	BYTE PUBLIC	
41	0039H	1	SOURCESTOP	BYTE	45
2			SPACE.	LITERALLY	
42	0019H	2	SSCIP.	WORD PUBLIC AT	
40	000BH	2	SSCIP.	WORD MEMBER(GLOB)	42
42	0027H	1	SSPEC.	BYTE PUBLIC AT	
40	0019H	1	SSPEC.	BYTE MEMBER(GLOB)	42
41	0066H	1	STOPTITLE.	BYTE	45

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		STKUC.	LITERALLY	43	
43	0002H	1 STYPE.	BYTE MEMBER(CURRENTSYMBOL)		
43	0002H	1 STYPE.	BYTE MEMBER(CURRENTSYMBOL)		
41	0047H	1 SYMBOLDEVICE	BYTE PUBLIC		
3		SYMBOLHEAD	LITERALLY		
3		SYMBOLSTRUC.	LITERALLY	43	
43	000AH	2 SYMTABADR	WORD PUBLIC		
4	0000H	2 T.	WORD PARAMETER	5	
2		TAB.	LITERALLY		
7	0000H	2 TEXTADR.	WORD PARAMETER	8	
2		THENDO	LITERALLY		
41	0048H	30 TITLE.	BYTE ARRAY(30) PUBLIC	48	
43	0002H	4 TOKEN.	STRUCTURE PUBLIC		
2		TRUE	LITERALLY		
43	0000H	1 TYPE	BYTE MEMBER(TOKEN)		
3		TYPEBIT.	LITERALLY		
3		TYPECOUNT.	LITERALLY		
25	0000H	UPPER.	PROCEDURE BYTE EXTERNAL(7) STACK=0000H		
43	0002H	2 VALUE.	WORD MEMBER(TOKEN)		

MODULE INFORMATION:

CODE AREA SIZE	= 002CH	44D
CONSTANT AREA SIZE	= 0000H	0D
VARIABLE AREA SIZE	= 0291H	657D
MAXIMUM STACK SIZE	= 000AH	10D
339 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ISIS-II PL/M-86 V2.0 COMPIATION OF MODULE C41

OBJECT MODULE PLACED IN CM.OBJ

COMPILER INVOKED BY: :F0: CM.PLN DEBUG OPTIMIZE(2) DATE(10/5/81) PAGELNGTH(100) XREF

```
1      $title ('codemacro module 1')
      cml:
      do;

      /*

      modified 7/24/81 R. Silberstein

      */

      /* This is the module to build new instructions
         which is not present in the already existing
         system. */

      $include (:f1:macro.lit)
= $nolist
      $include (:f1:equals.lit)
= $nolist
      $include (:f1:struc.lit)
= $nolist
      $include (:f1:cmacd.lit)
= $nolist
      $include (:f1:ermod.lit)
= $nolist
      $include (:f1:scan.ext)
= $nolist
      $include (:f1:ermod.ext)
= $nolist
      $include (:f1:cm2.ext)
= $nolist
      $include (:f1:cm.lit)
= $nolist
      $include (:f1:global.ext)
= $nolist
```

CCP/M-86 2.0 V.3

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104

\$eject

/* Subroutines: */

```

59 1  more$left$on$line: PROC byte;
60 2  if accum(0) <> cr then return true;
62 2  else return false;
63 2  end$proc more$left$on$line;

64 1  modrm$rout: PROC;
65 2  dcl nopar byte;
66 2  if token.type = number then$do
68 3  call put$b(mmodrm1);
69 3  if token.value > 7 then$do
71 4  cm$error=true; /* legal values are 0,1,...,7 */
72 4  return;
73 4  else$do
75 4  call put$b(token.value);
76 4  end$if;
77 3  else$do
79 3  if token.type = ident then$do
81 4  if legal$parameter(acclen,.accum(0),.nopar) then$do
83 5  call put$b(mmodrm2);
84 5  call put$b(nopar);
85 5  else$do /* error, parameter mismatch */
87 5  cm$error=true;
88 5  return;
89 5  end$if;
90 4  else$do /* error, expected parameter */
92 4  cm$error=true;
93 4  return;
94 4  end$if;
95 3  end$if;
96 2  call scan;
97 2  if accum(0) <> comma then$do
/* error, expected comma */
99 3  cm$error=true;
100 3  return;
101 3  end$if;
102 2  call scan;
103 2  if token.type = ident then$do
105 3  if legal$parameter(acclen,.accum(0),.nopar) then$do
107 4  call put$b(nopar);
108 4  call scan;
109 4  return;
110 4  else$do /* error, parameter mismatch */
112 4  cm$error=true;
113 4  return;
114 4  end$if;
115 3  else$do /* error, expected parameter */
117 3  cm$error=true;
118 3  return;
119 3  end$if;
120 2  end$proc modrm$rout;

121 1  db$dw$common$rout: PROC(directive);
122 2  dcl (directive,nopar) byte;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

123 2      if token.type = number then$do
125 3          call put$b(directive);
126 3          if directive = mdwn then$do
128 4              call put$w(token.value);
129 4          else$do
131 4              if token.value > 0fff then cm$error=true;
133 4                  else call put$b(token.value);
134 4          end$if;
135 3          call scan;
136 3          return;
137 3      else$do
139 3          if token.type = ident then$do
141 4              if legal$parameter(acclen,.accum(0),.nopar) then$do
143 5                  call put$b(directive+1);
144 5                  call put$b(nopar);
145 5                  call scan;
146 5                  return;
147 5              else$do /* error, parameter mismatch */
149 5                  cm$error=true;
150 5                  return;
151 5              end$if;
152 4              else$do /* error, expected parameter */
154 4                  cm$error=true;
155 4                  return;
156 4              end$if;
157 3          end$if;
158 2      end$proc db$dw$common$rou;

159 1      d$s$rb$rw$rou: PROC(directive);
160 2          dcl (directive,nopar) byte;
161 2          if token.type = ident then$do
163 3              if legal$parameter(acclen,.accum(0),.nopar) then$do
165 4                  call put$b(directive);
166 4                  call put$b(nopar);
167 4                  call scan;
168 4                  return;
169 4              else$do /* error, parameter mismatch */
171 4                  cm$error=true;
172 4                  return;
173 4              end$if;
174 3              else$do /* error, expected parameter */
176 3                  cm$error=true;
177 3                  return;
178 3              end$if;
179 2          end$proc d$s$rb$rw$rou;

180 1      nose$fix$rou: PROC;
181 2          dcl nopar byte;
182 2          call put$b(mnose$fix);
183 2          do case legal$seg$reg;
184 3              do; /* error, no segment register specified */
185 4                  cm$error=true;
186 4                  return;
187 4              end;
188 3              call put$b(res);
189 3              call put$b(rcs);
190 3              call put$b(rss);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
191 3      call put$b(rus);
192 3      end$case;
193 2      call scan;
194 2      if accum(0) <> comma then$do
        /* error, expected comma */
196 3      cm$error=true;
197 3      return;
198 3      end$if;
199 2      call scan; /* skip comma */
200 2      if legal$parameter(acclen,.accum(0),.nopar) then$do
202 3      call put$b(nopar);
203 3      call scan;
204 3      else$do
206 3      cm$error=true;
207 3      return;
208 3      end$if;
209 2      end$proc nose$fix$rout;

210 1      dbit$rout: PROC;
211 2      call put$b(mdbit);
212 2      end$proc dbit$rout;

213 1      field$descr$rout: PROC;
214 2      dcl nopar byte,cm$b$var based cmpt byte;
215 2      do forever;
216 3      if token.type <> number or token.value > 0fh then$do
        /* error, expected numberdef. */
218 4      cm$error=true;
219 4      return;
220 4      end$if;
221 3      call put$b(mnumberbits);
222 3      call put$b(token.value);
223 3      call scan;
224 3      if accum(0) <> leftpar then$do
        /* error, expected left parenthesis */
226 4      cm$error=true;
227 4      return;
228 4      end$if;
229 3      call scan; /* skip left parenthesis */
230 3      if token.type = ident then$do
232 4      if not legal$parameter(acclen,.accum(0),.nopar) then$do
        /* error, parameter mismatch */
234 5      cm$error=true;
235 5      return;
236 5      end$if;
237 4      cmpt=cmpt-2;
238 4      cm$b$var=cm$b$var-1; /* it was a parameter, not a number */
239 4      cmpt=cmpt+2;
240 4      call put$b(nopar);
241 4      call scan;
242 4      if accum(0) <> leftpar then$do
        /* error, expected left parenthesis */
244 5      cm$error=true;
245 5      return;
246 5      end$if;
247 4      call scan; /* skip left parenthesis */
248 4      end$if;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
249 3      if token.type <> number or token.value > 0 then $do
        /* error, expected numberdef.(byte) or parameter */
251 4          cm$error=true;
252 4          return;
253 4      end$if;
254 3      call put$b(token.value);
255 3      call scan;
256 3      if accum(0) <> rightpar then $do
        /* error, expected right paranthesis */
258 4          cm$error=true;
259 4          return;
260 4      end$if;
261 3      call scan; /* skip right paranthesis */
262 3      cmpt=cmpt-4;
263 3      if cm$b$var = mformalbits and
        accum(0) = rightpar then call scan;
265 3      cmpt=cmpt+4;
266 3      if accum(0) <> comma then return;
268 3      call scan;
269 3      end$forever;
270 2      end$proc field$descr$rout;

271 1      endd$bit$rout: PROC;
272 2          call put$b(mendbit);
273 2      end$proc endd$bit$rout;

274 1      endm$rout: PROC;
275 2          call put$b(menum);
276 2      end$proc endm$rout;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

\$eject

/* level 1 in the syntax-tree of codemacrobuilding */

```

277 1  COMMON$CM$ROUT: PROC (TYPE);
278 2  DECLARE TYPE BYTE;
279 2  IF pass = 1 THEN$do
281 3  CALL skip$rest$of$line;
282 3  RETURN;
283 3  END$IF;
284 2  CM$error=false;
285 2  IF NOT codemacro$flag THEN$do
    /* error, codemacro directive outside codemacrobodydef. */
287 3  CM$error=true;
288 3  END$IF;
289 2  DO CASE TYPE;
290 3  CALL db$dw$common$rouT(mdbn);
291 3  CALL db$dw$common$rouT(mdwn);
292 3  CALL d$s$rb$rw$rouT(mddf);
293 3  CALL d$s$rb$rw$rouT(ms$gfix);
294 3  CALL nose$gfix$rouT;
295 3  CALL modrm$rouT;
296 3  CALL d$s$rb$rw$rouT(mrelb);
297 3  CALL d$s$rb$rw$rouT(mrelw);
298 3  DO;
299 4  CALL dbit$rouT;
300 4  CALL field$desc$rouT;
301 4  CALL enddbit$rouT;
302 4  END;
303 3  END$CASE;
304 2  IF CM$error OR more$left$on$line THEN$do
    /* error */
306 3  global$CM$error=true;
307 3  CALL errmsg(codemacroerr);
308 3  END$IF;
309 2  CALL skip$rest$of$line;
310 2  END COMMON$CM$ROUT;

311 1  codemacro$rouT: PROC public;
312 2  IF pass = 1 THEN$do
314 3  codemacro$flag=true;
315 3  CALL skip$rest$of$line;
316 3  RETURN;
317 3  END$IF;
318 2  CM$error=false;
319 2  global$CM$error=false;
320 2  IF codemacro$flag THEN$do
    /* error, nested codemacrodefinition */
322 3  CM$error=true;
323 3  END$IF;
324 2  CALL init$CM$rouT; /* clearing all temp. working storages */
325 2  codemacro$flag=true;
326 2  IF NOT name$rouT THEN$do
    /* error, expected codemacroname */
328 3  CM$error=true;
329 3  END$IF;
330 2  CALL formal$list$rouT;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____


```

331 2      if cm$error or more$left$on$line; then$do
          /* error */
333 3      global$cm$error=true;
334 3      call errmsg(codemacroerr);
335 3      end$if;
336 2      call skip$rest$of$line;
337 2      end$proc codemacro$rout;

338 1      db$cm$rout: PROC public;
339 2      CALL COMMON$CM$ROUT (0);
340 2      end$proc db$cm$rout;

341 1      dw$cm$rout: PROC public;
342 2      CALL COMMON$CM$ROUT (1);
343 2      end$proc dw$cm$rout;

344 1      dd$cm$rout: PROC public;
345 2      CALL COMMON$CM$ROUT (2);
346 2      end$proc dd$cm$rout;

347 1      segfix$cm$rout: PROC public;
348 2      CALL COMMON$CM$ROUT (3);
349 2      end$proc segfix$cm$rout;

350 1      nose$fix$cm$rout: PROC public;
351 2      CALL COMMON$CM$ROUT (4);
352 2      end$proc nose$fix$cm$rout;

353 1      modrm$cm$rout: PROC public;
354 2      CALL COMMON$CM$ROUT (5);
355 2      end$proc modrm$cm$rout;

356 1      relb$cm$rout: PROC public;
357 2      CALL COMMON$CM$ROUT (6);
358 2      end$proc relb$cm$rout;

359 1      relw$cm$rout: PROC public;
360 2      CALL COMMON$CM$ROUT (7);
361 2      end$proc relw$cm$rout;

362 1      dbit$cm$rout: PROC public;
363 2      CALL COMMON$CM$ROUT (8);
364 2      end$proc dbit$cm$rout;

365 1      end$cm$rout: PROC public;
366 2      if pass = 1 then$do
368 3      call skip$rest$of$line;
369 3      codemacro$flag=false;
370 3      return;
371 3      end$if;
372 2      cm$error=false;
373 2      if not codemacro$flag then$do
          /* error, terminating a not yet started codemacro */
375 3      cm$error=true;
376 3      end$if;
377 2      call endm$rout;
378 2      if pass = 0 then call terminate$cm$rout;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```
380 2      if cm$error or moreleftthan$line or callistoverflow then do
          /* error */
          call errmsg(codemacroerr);
382 3      end$if;
383 3      call skip$rest$of$line;
384 2      global$cm$error=false;
385 2      codemacro$flag=false;
386 2      end$proc end$cm$rout;
387 2
388 1      end$module cm1;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
56	0000H	4	APSADDR. BYTE ARRAY(4) EXTERNAL(51)
56	0000H	1	ACCLEN BYTE EXTERNAL(55) 81 105 141 163 200
			232
56	0000H	1	ACCLNSAVE BYTE EXTERNAL(58)
56	0000H	80	ACCUMS BYTE ARRAY(80) EXTERNAL(56) 60 81 97
			105 141 163 194 200 224 232 242 256 263
			266
56	0000H	80	ACCUMSAVE. BYTE ARRAY(80) EXTERNAL(57)
2			ADDR LITERALLY 56
49	0000H	1	b. BYTE PARAMETER 50
56	0008H	1	BASEINDEX. BYTE MEMBER(OPERANDS)
56	0008H	1	BASEINDEX. BYTE MEMBER(CURRENTSYMBOL)
8			BASEREGBIT LITERALLY
8			BASEREGCOUNT LITERALLY
8			BREGBIT. LITERALLY
8			BREGCOUNT. LITERALLY
4			BYT. LITERALLY
56	0000H	2	CIP. WORD EXTERNAL(20)
1	0000H		CM1. PROCEDURE STACK=0000H
214	0000H	1	CMVAR BYTE BASE(CMPT) 238 263
35	0000H	1	CMERROR. BYTE EXTERNAL(7) 71 87 92 99 112
			117 132 149 154 171 176 185 196 206 218
			226 234 244 251 258 284 287 304 318 322
			328 331 372 375 380
35	0000H	1	CMLISTOVERFLOW BYTE EXTERNAL(9) 380
35	0000H	2	CMPT. WORD EXTERNAL(10) 214 237 238 239 262
			263 265
3			CODE LITERALLY
17			CODEMACROERR LITERALLY 307 334 382
56	0000H	1	CODEMACROFLAG. BYTE EXTERNAL(77) 285 314 320 325 369
			373 386
8			CODEMACROHEAD. LITERALLY
56	0000H	2	CODEMACROPTR WORD EXTERNAL(72)
311	0313H	91	CODEMACROROUT. PROCEDURE PUBLIC STACK=0006H
55			COLON. LITERALLY
55			COMMA. LITERALLY 97 194 266
277	0286H	141	COMMONCMROUT PROCEDURE STACK=0012H 339 342 345 348
			351 354 357 360 363
2			CR LITERALLY 60
56	0000H	2	CSCIP. WORD EXTERNAL(25)
56	0000H	1	CSEGSPEC BYTE EXTERNAL(23)
56	0000H	1	CSEGTTYPE BYTE EXTERNAL(21)
56	0000H	2	CSEGVALUE. WORD EXTERNAL(22)
12			CSDVER LITERALLY
56	0000H	1	CSPEC. BYTE EXTERNAL(32)
56	0000H	2	CURCSEG. WORD EXTERNAL(28)
56	0000H	2	CURDSEG. WORD EXTERNAL(29)
56	0000H	2	CURESEG. WORD EXTERNAL(31)
56	0000H	9	CURRENTSYMBOL. STRUCTURE EXTERNAL(68)

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER # _____

56	0000H	2	CURRSEG.	WORD EXTERNAL(30)				
338	036EH	11	DBCMROUT	PROCEDURE PUBLIC STACK=0016H				
121	0092H	108	DBDWCOMMONROUT	PROCEDURE STACK=000CH	290	291		
362	03C6H	11	DBITCMROUT	PROCEDURE PUBLIC STACK=0016H				
210	019EH	11	DBITROUT	PROCEDURE STACK=0006H	299			
2			DCL.	LITERALLY				
344	0384H	11	DDCMROUT	PROCEDURE PUBLIC STACK=0016H				
56	0000H	1	DEFAULTORIVE	BYTE EXTERNAL(75)				
3			DELETEDSYMB.	LITERALLY				
56	0001H	1	DESCR.	BYTE MEMBER(TOKEN)				
121	0004H	1	DIRECTIVE.	BYTE PARAMETER AUTOMATIC	122	125	126	143
159	0004H	1	DIRECTIVE.	BYTE PARAMETER AUTOMATIC	160	165		
3			DOUBLEDEFINED.	LITERALLY				
17			DOUBLEDEFVAR	LITERALLY				
17			DOUBLEDEFSYMB.	LITERALLY				
17			DOUBLEDEFVAR	LITERALLY				
16			DOUBLERANGE.	LITERALLY				
56	0000H	2	DSCIP.	WORD EXTERNAL(27)				
12			DSOVER	LITERALLY				
56	0000H	1	DSPEC.	BYTE EXTERNAL(33)				
159	00FEH	56	DSRBRWRROUT	PROCEDURE STACK=000CH	292	293	296	297
341	0379H	11	DWCMROUT	PROCEDURE PUBLIC STACK=0016H				
4			DWRD	LITERALLY				
2			ELSEDD	LITERALLY				
30	0000H		EMPTYLINE.	PROCEDURE BYTE EXTERNAL(5) STACK=0000H				
2			ENDCASE.	LITERALLY				
365	03D1H	83	ENDCMROUT.	PROCEDURE PUBLIC STACK=000AH				
271	0270H	11	ENDDBTROUT.	PROCEDURE STACK=0006H	301			
2			ENDDO.	LITERALLY				
2			ENDFOREVER	LITERALLY				
2			ENDIF.	LITERALLY				
2			ENDMODULE.	LITERALLY				
274	0278H	11	ENDMROUT	PROCEDURE STACK=0006H	377			
2			ENDOFFILE.	LITERALLY				
17			ENDOFFLINEERR	LITERALLY				
2			ENDPROC.	LITERALLY				
2			ENDWHILE	LITERALLY				
56	0000H	1	EOFSET	BYTE EXTERNAL(59)				
32	0000H		ERRMSG	PROCEDURE EXTERNAL(6) STACK=0000H	307	334		
32	0000H	1	ERRNO.	BYTE PARAMETER 33				
3			ERROR.	LITERALLY				
56	0000H	1	ERRORPKINTED	BYTE EXTERNAL(62)				
56	0000H	2	ERRORS	WORD EXTERNAL(52)				
56	0000H	2	ESCIP.	WORD EXTERNAL(24)				
12			ESOVER	LITERALLY				
56	0000H	1	ESPEC.	BYTE EXTERNAL(35)				
2			FALSE.	LITERALLY	62	284	318	319 369 372 385
213	01A9H	199	FIELDDESCRROUT	PROCEDURE STACK=000AH	300			
8			FILEISTRUCTURE	LITERALLY				
8			FILEQSTRUCTURE	LITERALLY				
17			FILESYNTERR.	LITERALLY				
55			FIRST.	LITERALLY				
17			FIRSTITEM.	LITERALLY				
56	0000H	2	FIRSTHACROPTR.	WORD EXTERNAL(63)				
2			FOREVER.	LITERALLY	215			

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

40	0000H	FORMALLISTROUT . . .	PROCEDURE EXTERNAL(13) STACK=0000H	340
2		FORMFEED	LITERALLY	
56	0000H	1 FULLSYMBTAG	BYTE EXTERNAL(65)	
35	0000H	1 GLOBALCHKERR	BYTE EXTERNAL(8)	306 319 333 385
57	0000H	GLOBALINIT	PROCEDURE EXTERNAL(78) STACK=0000H	
56	0000H	5 HELP	BYTE ARRAY(5) EXTERNAL(73)	
56	0000H	1 L	BYTE EXTERNAL(74)	
3		IDENT	LITERALLY	79 103 139 161 230
56	0000H	1 IFLEVEL	BYTE EXTERNAL(67)	
56	0000H	1 IFLIST	BYTE EXTERNAL(38)	
17		IFPARERR	LITERALLY	
17		ILLEGALMACRO	LITERALLY	
17		ILLEGALPSEUDO	LITERALLY	
17		ILLEXPRELEM	LITERALLY	
17		ILLTOPER	LITERALLY	
56	0000H	1 INCLUDEDDEFAULT . . .	BYTE EXTERNAL(76)	
56	0000H	1 INCLUDEON	BYTE EXTERNAL(66)	
8		INDEXREGBIT	LITERALLY	
8		INDEXREGCOUNT	LITERALLY	
2		INIT	LITERALLY	
36	0000H	INITCMROUT	PROCEDURE EXTERNAL(11) STACK=0000H	324
17		INSTRERR	LITERALLY	
56	0000H	1 INTELHEXON	BYTE EXTERNAL(60)	
8		IREGBIT	LITERALLY	
8		IREGCOUNT	LITERALLY	
3		LAB	LITERALLY	
17		LABOUTOFRANGE	LITERALLY	
7		LEFTBRACKET	LITERALLY	
55		LEFTPARE	LITERALLY	224 242
44	0000H	LEGALPARAMETER	PROCEDURE BYTE EXTERNAL(15) STACK=0000H	81
			105 141 163 200 232	
47	0000H	LEGALSEGREG	PROCEDURE BYTE EXTERNAL(16) STACK=0000H	183
56	0000H	2 LENGTH	WORD MEMBER(CURRENTSYMBOL)	
56	0000H	2 LENGTH	WORD MEMBER(OPERANDS)	
2		LF	LITERALLY	
44	0000H	1 LG	BYTE PARAMETER	45
2		LIT	LITERALLY	2 3 4 5 6 7 8
			9 10 11 12 13 14 15 16 17 55	
56	0000H	2 MACROPTR	WORD EXTERNAL(64)	
56	0000H	1 MAXCOL	BYTE EXTERNAL(39)	
9		MDBF	LITERALLY	
9		MDBIT	LITERALLY	211
9		MDBN	LITERALLY	290
9		MDDF	LITERALLY	292
9		MDFW	LITERALLY	
9		MDWN	LITERALLY	126 291
9		MENDBIT	LITERALLY	272
9		MENDM	LITERALLY	275
9		MFORMALBITS	LITERALLY	263
17		MISSIFERR	LITERALLY	
17		MISSINGPSEUDO	LITERALLY	
17		MISSINSTR	LITERALLY	
17		MISSSEGMINFO	LITERALLY	
17		MISSYPEMINFO	LITERALLY	
9		MMODRM1	LITERALLY	68
9		MMODRM2	LITERALLY	83

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

9			MNOSEGFIX.	LITERALLY	182	
9			MNUMBERBITS.	LITERALLY	221	
11			MODB.	LITERALLY		
11			MODD.	LITERALLY		
15			MODLETTERBIT.	LITERALLY		
15			MODLETTERCOUNT.	LITERALLY		
353	03A5H	11	MODRMCMROUT.	PROCEDURE PUBLIC STACK=0016H		
64	0012H	128	MODRMROUT.	PROCEDURE STACK=000AH	295	
11			MODSB.	LITERALLY		
11			MODW.	LITERALLY		
59	0000H	18	MORELEFTONLINE.	PROCEDURE BYTE STACK=0002H	304 331 380	
9			MRELB.	LITERALLY	296	
9			MRELW.	LITERALLY	297	
9			MSEGFIX.	LITERALLY	293	
38	0000H		NAMEROUT.	PROCEDURE BYTE EXTERNAL(12) STACK=0000H	326	
3			NEGLECTED.	LITERALLY		
17			NEGLECTERR.	LITERALLY		
17			NESTEDIFERR.	LITERALLY		
17			NESTEDINCLUDEERR.	LITERALLY		
56	0000H	1	NEXTCH.	BYTE EXTERNAL(54)		
4			NIL.	LITERALLY		
56	0000H	1	NOERROR.	BYTE EXTERNAL(61)		
11			NOMOD.	LITERALLY		
56	0000H	1	NOOPER.	BYTE EXTERNAL(70)		
8			NOOVERCOUNT.	LITERALLY		
8			NOOVERRIDEBIT.	LITERALLY		
214	0004H	1	NOPAR.	BYTE	232 240	
181	0003H	1	NOPAR.	BYTE	200 202	
160	0002H	1	NOPAR.	BYTE	163 166	
122	0001H	1	NOPAR.	BYTE	141 144	
65	0000H	1	NOPAR.	BYTE	81 94 105 107	
13			NOPARAND.	LITERALLY		
16			NORANGE.	LITERALLY		
350	039AH	11	NOSEGFIXCMROUT.	PROCEDURE PUBLIC STACK=0016H		
180	0136H	104	NOSEGFIXROUT.	PROCEDURE STACK=000AH	294	
3			NUMBER.	LITERALLY	66 123 216 249	
16			NUMBERRANGE.	LITERALLY		
7			OAND.	LITERALLY		
7			OEQ.	LITERALLY		
56	0006H	2	OFFSET.	WORD MEMBER(OPERANDS)		
56	0006H	2	OFFSET.	WORD MEMBER(CURRENTSYMBOL)		
7			OGE.	LITERALLY		
7			OGT.	LITERALLY		
7			OLAST.	LITERALLY		
7			OLE.	LITERALLY		
7			OLENGTH.	LITERALLY		
7			OLT.	LITERALLY		
7			OMOD.	LITERALLY		
7			ONE.	LITERALLY		
7			ONOT.	LITERALLY		
7			OOFFSET.	LITERALLY		
7			ODR.	LITERALLY		
56	0000H	36	OPERANDS.	STRUCTURE ARRAY(4) EXTERNAL(71)		
8			OPERANDSTRUC.	LITERALLY	56	
3			OPERATOR.	LITERALLY		
17			OPHSMATCH.	LITERALLY		
7			OPTR.	LITERALLY		

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

7			USEG	LITERALLY	
7			OSHL	LITERALLY	
7			OSHORT	LITERALLY	
7			OSHR	LITERALLY	
7			OTYPE	LITERALLY	
7			OXOR	LITERALLY	
56	0000H	1	PAGESIZE	BYTE EXTERNAL(45)	
56	0000H	1	PASS	BYTE EXTERNAL(19)	279 312 356 378
6			PCOPLMACRO	LITERALLY	
6			PCSEG	LITERALLY	
6			PDB	LITERALLY	
6			PDBIT	LITERALLY	
6			PDD	LITERALLY	
6			PDSEG	LITERALLY	
6			PDW	LITERALLY	
6			PEJECT	LITERALLY	
6			PEHD	LITERALLY	
6			PENDIF	LITERALLY	
6			PENDM	LITERALLY	
6			PEQU	LITERALLY	
6			PESEG	LITERALLY	
6			PIF	LITERALLY	
6			PIFLIST	LITERALLY	
6			PINCLUDE	LITERALLY	
6			PLIST	LITERALLY	
6			PHOORM	LITERALLY	
6			PNOIFLIST	LITERALLY	
6			PNOLIST	LITERALLY	
6			PNOSEGFTX	LITERALLY	
6			PORG	LITERALLY	
6			PPAGESIZE	LITERALLY	
6			PPAGEWIDTH	LITERALLY	
6			PRB	LITERALLY	
56	0000H	240	PREFIX	BYTE ARRAY(240) EXTERNAL(49)	
13			PREFIXAND	LITERALLY	
14			PREFIXON	LITERALLY	
56	0000H	1	PREFIXPTR	BYTE EXTERNAL(50)	
6			PRELB	LITERALLY	
6			PRELW	LITERALLY	
56	0000H	1	PRINTDEVICE	BYTE EXTERNAL(42)	
56	0000H	1	PRINTON	BYTE EXTERNAL(36)	
56	0000H	1	PRINTSWITCHOFF	BYTE EXTERNAL(37)	
2			PROC	LITERALLY	18 20 22 24 27 30 32
					36 38 40 42 44 47 49 52 59 64
					121 159 180 210 213 271 274 277 311 338
					341 344 347 350 353 356 359 362 365
6			PRS	LITERALLY	
6			PRW	LITERALLY	
6			PSEGFTX	LITERALLY	
3			PSEUDO	LITERALLY	
17			PSEUDOOPERK	LITERALLY	
6			PSIFORM	LITERALLY	
6			PSSEG	LITERALLY	
6			PTITLE	LITERALLY	
44	0000H	2	PTR	WORD PARAMETER	45
44	0000H	2	PTR2	WORD PARAMETER	45
49	0000H		PUTB	PROCEDURE EXTERNAL(17) STACK=0000H	68 75

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

			33	84	107	125	133	143	144	155	166	182
			188	189	190	191	202	211	221	222	240	254
			272	275								
52	0000H		PUTW									
5			RAH.									
5			RAL.									
16			RANGEAND									
15			RANGESPECCBIT									
15			RANGESPECCOUNT									
16			RANGETYPEAND									
5			RAX.									
5			RBH.									
5			RBL.									
5			RBP.									
5			RBX.									
5			RCH.									
5			RCL.									
5			RCS.									
5			RCX.									
5			RDH.									
5			RDI.									
5			RDL.									
5			RDS.									
5			RDX.									
2			REENT.									
5			REG.									
16			REGISTERRANGE.									
356	03B0H	11	RELBCROUT									
359	03B8H	11	RELWCROUT									
5			RES.									
7			RIGHTBRACKET									
55			RIGHTPAR									
5			RSI.									
5			RSP.									
5			RSS.									
56	0000H	12	SAVESOURCE									
20	0000H		SCAN									
18	0000H		SCANINIT									
55			SECOND									
347	03BFH	11	SEGFIXCROUT									
8			SEGMBIT.									
8			SEGMCOUNT.									
56	0004H	2	SEGMENT.									
56	0004H	2	SEGMENT.									
8			SEGTYPEBIT									
8			SEGTYPECJUNT									
56	0003H	1	SFLAG.									
56	0003H	1	SFLAG.									
56	0000H	1	SIMFORM.									
16			SINGLERANGE.									
22	0000H		SKIPRESTOFLINE									
27	0000H		SKIPUNTIL.									
56	0000H	80	SOURCEBUF.									
56	0000H	12	SOURCENAME									

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. #


```

56 0000H 1 SOURCEPTR. . . . . BYTE EXTERNAL(48)
2 SPACE. . . . . LITERALLY
3 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
24 0000H SPECIALTOKEN. . . . . PROCEDURE BYTE EXTERNAL(3) STACK=0000H
10 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
10 SPEC. . . . . LITERALLY
56 0000H 2 SSCIP. . . . . WORD EXTERNAL(26)
12 SSOVER. . . . . LITERALLY
56 0000H 1 SSPEC. . . . . BYTE EXTERNAL(34)
3 STRING. . . . . LITERALLY
2 STRUC. . . . . LITERALLY 56
56 0002H 1 STYPE. . . . . BYTE MEMBER(OPERANDS)
56 0002H 1 STYPE. . . . . BYTE MEMBER(CURRENTSYMBOL)
3 SYMBOL. . . . . LITERALLY
56 0000H 1 SYMBOLDEVICE. . . . . BYTE EXTERNAL(43)
8 SYMBOLHEAD. . . . . LITERALLY
8 SYMBOLSTRUC. . . . . LITERALLY 56
56 0000H 2 SYMBTABADR. . . . . WORD EXTERNAL(69)
2 TAB. . . . . LITERALLY
42 0000H TERMINATECMROUT. . . . . PROCEDURE EXTERNAL(14) STACK=0000H 379
2 THENDO. . . . . LITERALLY 66 69 79 81 97 103 105
123 126 139 141 161 163 194 200 216 224
230 232 242 249 256 279 285 304 312 320
326 331 366 373 380
56 0000H 30 TITLE. . . . . BYTE ARRAY(30) EXTERNAL(44)
27 0000H 1 TOK. . . . . BYTE PARAMETER 28
24 0000H 1 TOK. . . . . BYTE PARAMETER 25
56 0000H 4 TOKEN. . . . . STRUCTURE EXTERNAL(53) 66 69 75 79
103 123 128 131 133 139 161 216 222 230
249 254
2 TRUE. . . . . LITERALLY 61 71 87 92 99 112 117
132 149 154 171 176 185 196 206 215 218
226 234 244 251 258 287 306 314 322 325
328 333 375
277 0004H 1 TYPE. . . . . BYTE PARAMETER AUTOMATIC 278 289
56 0000H 1 TYPE. . . . . BYTE MEMBER(TOKEN) 66 79 103 123 139
161 216 230 249
8 TYPEBIT. . . . . LITERALLY
8 TYPECOUNT. . . . . LITERALLY
3 UDEFSYMB. . . . . LITERALLY
17 UDEFSYMBOL. . . . . LITERALLY
56 0002H 2 VALUE. . . . . WORD MEMBER(TOKEN) 69 75 128 131 133
216 222 249 254
3 VARIABLE. . . . . LITERALLY
52 0000H 2 W. . . . . WORD PARAMETER 53
4 WRD. . . . . LITERALLY

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CODE AREA SIZE = 0424H 10600
CONSTANT AREA SIZE = 0000H 00
VARIABLE AREA SIZE = 0005H 50
MAXIMUM STACK SIZE = 0016H 220
936 LINES READ
0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

ISIS-11 PL/M-86 V2.0 COMPILATION OF MODULE CM2

OBJECT MODULE PLACED IN CM2.OBJ

COMPILER INVOKED BY: :F0: CM2.PLN DEBUG OPTIMIZE(2) DATE(10/5/81) PAGEWIDTH(100) XREF

1 \$title ("codemacro module 2")

cm2:

do;

/*

modified 3/26/81 R. Silberstein

*/

/* This is the module to build new instructions
which is not present in the already existing
system. */

```
= $include (:f1:macro.lit)
= $nolist
= $include (:f1:equals.lit)
= $nolist
= $include (:f1:struc.lit)
= $nolist
= $include (:f1:cmacd.lit)
= $nolist
= $include (:f1:cm.lit)
= $nolist
= $include (:f1:symb.ext)
= $nolist
= $include (:f1:subr1.ext)
= $nolist
= $include (:f1:subr2.ext)
= $nolist
= $include (:f1:scan.ext)
= $nolist
= $include (:f1:files.ext)
= $nolist
= $include (:f1:exglob.ext)
= $nolist
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$object

/* DECLARATION FOR "CODEMACRO" PART

Extended version of ASM86 */

```

148 1  dcl    cm$name(80)      byte           ,
        codemacro$found      byte           ,
        cm$in$pt             address         ,
        cm$in$var            based          cm$in$pt byte,
        cm$in$lg             byte           ,
        par$name(80)         byte           ,
        pmpt                 address         ,
        cm$pm$var            based          pmpt   byte,
        cm$error             byte           public,
        cm$body(100)         byte           ,
        cmpt                 address         public,
        cm$b$var            based          cmpt   byte,
        cm$w$var            based          cmpt   addr,
        cm$counter           addr           ,
        global$cm$error      byte           public,
        cm$body$full         byte           ,
        ant$par              byte           ,
        cm$list$overflow     byte           public;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
$eject
```

```
/*  VARIOUS SUBROUTINES */
```

```

149 1  legal$parameter: PROC(lg,ptr,ptr2) byte public;
150 2  dcl (lg,i) byte,
      (ptr,ptr2) address,
      no based ptr2 byte;

151 2  i=0;
152 2  do no=0 to cm$body(?) - 1;
153 3  if parname(i+lg) = 0 and
      equal(lg,ptr,par$name(i)) then return true;
155 3  do while par$name(i:=i+1) <> 0;
156 4  endwhile;
157 3  i=i+1;
158 3  end;
159 2  return false;
160 2  end$proc legal$parameter;

161 1  legal$spec$letter: PROC(1) byte;
162 2  dcl (1,i) byte;
163 2  dcl table(8) byte data ("ACDENRSX");
164 2  do i=0 to last(table);
165 3  if table(i) = 1 then return i;
167 3  end;
168 2  return i;
169 2  end$proc legal$spec$letter;

170 1  legal$mod$letter: PROC(1) byte;
171 2  dcl (1,i) byte;
172 2  dcl table(4) byte data ("B4DS");
173 2  do i=0 to last(table);
174 3  if table(i) = 1 and i < 3 then return i;
176 3  if table(i) = 1 and i = 3 then$do
178 4  if accum(2) = "B" then return 3;
      end$if;
181 3  end;
182 2  return i;
183 2  end$proc legal$mod$letter;

184 1  legal$register: PROC byte;
185 2  declare disp byte;
186 2  if token.type <> reg then return 0;
188 2  disp=0;
189 2  if token.descr=byt then disp=8;
191 2  if token.descr=dword then disp=16;
193 2  return token.value + disp + 1;
194 2  end$proc legal$register;

195 1  legal$seg$reg: PROC byte public;
196 2  if token.type <> reg then return 0;
198 2  if token.descr <> dword then return 0;
200 2  return token.value + 1;
201 2  end$proc legal$seg$reg;

202 1  put$b: PROC(b) public;
203 2  dcl b byte;

```

CCP/M-86 2.0 V.3
 COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # CC-104

```
204 2      cm$counter=cm$counter+1;
205 2      if cm$counter > 99 then$do
207 3          cm$error=true;
208 3          cm$body$full=true;
209 3          return;
210 3      end$if;
211 2      cm$b$var=b;
212 2      cmpt=cmpt+1;
213 2      end$proc put$b;

214 1      put$w: PROC(w) public;
215 2          dcl w addr;
216 2          cm$counter=cm$counter+2;
217 2          if cm$counter > 99 then$do
219 3              cm$error=true;
220 3              cm$body$full=true;
221 3              return;
222 3          end$if;
223 2          cm$w$var=w;
224 2          cmpt=cmpt+2;
225 2          end$proc put$w;

226 1      update$cm$lists: PROC byte;
227 2          dcl listptr address,next based listptr address;
228 2          dcl ptr address;
229 2          ptr=.listptr;
230 2          if findcodemacro(cm$n$lg,.cm$name(0),ptr)
232 3              then$do
233 4                  do while next <> 0;
234 4                      listptr=next;
235 4                      end$while;
236 3              else$do
237 3                  if not new$cm$name(cm$n$lg,.cm$name(0),ptr)
239 4                      then return false;
240 3                  end$if;
241 2                  next=freept;
242 2                  if not new$cm$body(cm$counter,.cm$body(0))
243 3                      then return false;
244 2                  return true;
245 2          end$proc update$cm$lists;
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$eject

/* level 4 in the syntax-tree of coeemacro building */

```

245 1 register$rou: PROC (1);
246 2     dcl 1 byte;
247 2     call put$b(1);
248 2 end$proc register$rou;

249 1 range$rou: PROC;

250 2     put$range: PROC(time);
251 3     dcl time byte;
252 3     if token.type = number then$do
254 4         if time = first then$do
256 5             cm$b$var=cm$b$var+numberrange;
257 5             cmpt=cmpt+1;
258 5         end$if;
259 4         if token.value > 0fff then$do
261 5             cm$error=true; /* too large number */
262 5             return;
263 5         else$do
265 5             call put$b(token.value);
266 5         end$if;
267 4     else$do
269 4         if time = first then$do
271 5             cm$b$var=cm$b$var+registerrange;
272 5             cmpt=cmpt+1;
273 5         end$if;
274 4         do case legal$register;
275 5             do; /* error, expecting a register spec. */
276 6                 cm$error=true;
277 6                 return;
278 6             and;
279 5                 call register$rou(rax);
280 5                 call register$rou(rcx);
281 5                 call register$rou(rdx);
282 5                 call register$rou(rbx);
283 5                 call register$rou(rsp);
284 5                 call register$rou(rbp);
285 5                 call register$rou(rsi);
286 5                 call register$rou(rdi);
287 5                 call register$rou(ral);
288 5                 call register$rou(rc1);
289 5                 call register$rou(rdl);
290 5                 call register$rou(rol);
291 5                 call register$rou(rah);
292 5                 call register$rou(rch);
293 5                 call register$rou(rdh);
294 5                 call register$rou(rbh);
295 5                 call register$rou(res);
296 5                 call register$rou(rcs);
297 5                 call register$rou(rss);
298 5                 call register$rou(rds);
299 5             end$case;
300 4         end$if;
301 3     call scan;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```
302 3      end$proc put$range;

303 2      s$range: PROC;
304 3          cmpt=cmpt-1;
305 3          cm$b$var=cm$b$var+singlerange;
306 3          call put$range(first);
307 3      end$proc s$range;

308 2      d$range: PROC;
309 3          cmpt=cmpt-1;
310 3          cm$b$var=cm$b$var+doublerange;
311 3          call put$range(first);
312 3          if accum(0) <> comma then$do
314 4              cm$error=true;
315 4              return;
316 4          end$if;
317 3          call scan;
318 3          call put$range(second);
319 3          if accum(0) <> rightpar then$do
321 4              cm$error=true;
322 4              return;
323 4          end$if;
          end$proc d$range;

/* mainpart of range routine */

325 2      call scan; /* skip left paranthesis */
326 2      if nextch = comma then call d$range;
328 2          else call s$range;
329 2      call scan;
330 2      end$proc range$rout;

331 1      spec$letter$rout: PROC(1);
332 2          dcl 1 byte;
333 2          call put$b(1);
334 2      end$proc spec$letter$rout;

335 1      mod$letter$rout: PROC(1);
336 2          dcl 1 byte;
337 2          call put$b(1);
338 2      end$proc mod$letter$rout;
```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

\$reject

/* level 3 in the syntax-tree of codemacro building */

```

339 1  par$descr$rout: PROC;
340 2      call copy(acclen,.accum(0),.cm$pm$var);
341 2      pmpt=pmpt+acclen;
342 2      cm$pm$var=0; /* end of par.name */
343 2      pmpt=pmpt+1;
344 2      call scan;
345 2      if accum(0) <> colon then$do
          /* error, expected colon in parameterdecl */
347 3          cm$error=true;
348 3      end$if;
349 2      call scan;
350 2      do case legal$spec$letter(accum(0));
351 3          call spec$letter$rout(speca);
352 3          call spec$letter$rout(specb);
353 3          call spec$letter$rout(specd);
354 3          call spec$letter$rout(specf);
355 3          call spec$letter$rout(specm);
356 3          call spec$letter$rout(specr);
357 3          call spec$letter$rout(specs);
358 3          call spec$letter$rout(specx);
359 3          do; /* error, expected specifier letter */
360 4              cm$error=true;
361 4              return;
362 4          end;
363 3      end$case;

364 2      do case legal$mod$letter(accum(1));
365 3          call mod$letter$rout(modb);
366 3          call mod$letter$rout(modw);
367 3          call mod$letter$rout(modd);
368 3          call mod$letter$rout(modsb);
369 3          call mod$letter$rout(nomod); /* no modletter */
370 3      end$case;
371 2      call scan;
372 2      if accum(0) = leftpar then call range$rout;
374 2      cm$body(2)=cm$body(2)+1;
375 2  end$proc par$descr$rout;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

\$eject

/* level 2 in the syntax-tree of codemacro building, */

/* Procedure to initialize temporary storage and pointers
concerning the building of codemacro */

```

376 1  init$cm$rout: PROC public;
377 2      cm$in$pt=.cm$name(0);
378 2      cmpt=.cm$body(0)+3; /* correcting for the head */
379 2      papt=.par$name(0);
380 2      CALL FILL (0, LENGTH (CM$NAME), .CM$NAME);
381 2      CALL FILL (0, LENGTH (CM$BODY), .CM$BODY);
382 2      CALL FILL (0, LENGTH (PAR$NAME), .PAR$NAME);
383 2      ant$par=0;
384 2      cm$counter=3;
385 2      cm$body$full=false;
386 2      cm$list$overflow=false;
387 2  end$proc init$cm$rout;

388 1  name$rout: PROC public;
389 2      if token.type <> ident then return false;
391 2      call copy(acclen,.accum(0),.cm$name(0));
392 2      cm$in$lg=acclen;
393 2      call scan;
394 2      return true;
395 2  end$proc name$rout;

396 1  formal$list$rout: PROC public;
397 2      do while token.type = ident;
398 3          call par$descr$rout;
399 3          if accum(0) <> ',' then return; /* end of parameters */
401 3          call scan;
402 3          end$while;
403 2  end$proc formal$list$rout;

404 1  terminate$cm$rout: PROC public;
405 2      if global$cm$error then$do
          /* error present in codemacrodef, */
          /* no updating of codemacrolist */
407 3          return;
408 3      end$if;
409 2      if not update$cm$lists
          then$do
411 3          cm$error=true; /* overflow, no more vacant memory */
412 3          cm$list$overflow=true;
413 3          return;
414 3      end$if;
      end$proc terminate$cm$rout;

416 1  end$module cm2;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
147	0000H	1	ACCLEN
147	0000H	80	ACCUM.
2			ADDR
72	0000H		ALPHANUMERIC
148	0115H	1	ANTPAR
75	0000H		ASCTICAR.
45	0000H	2	ASCTIPTR
202	0004H	1	8.
147	0008H	1	BASEINDEX.
8			BASEREGBIT
8			BASEREGCOUNT
8			BREGBIT.
8			BREGCOUNT.
4			BYT.
69	0000H	1	CH
72	0000H	1	CH
75	0000H	1	CH
63	0000H	1	CH
110	0000H	1	CH
78	0000H	1	CH
113	0000H	1	CH
66	0000H	1	CH
107	0000H	1	CH
147	0000H	2	CIP.
136	0000H		CLOSEHEX
134	0000H		CLOSEINCLUDE
138	0000H		CLOSEPRINT
130	0000H		CLOSESOURCE.
140	0000H		CLOSESYMBOL.
1	0000H		CM2.
148	00AFH	100	CMBODY
148	0114H	1	CMBODYFULL
148	0000H	1	CMVAR
148	0006H	2	CMCOUNTER.
148	00AEH	1	CMERROR.
148	0116H	1	CMLISTOVERFLOW
148	000CH	80	CMNAME
148	005DH	1	CMNLG.
148	0000H	2	CMNPT.
148	0000H	1	CMNVAR
148	0000H	1	CMPMVAR.
148	0004H	2	CMPT
148	0000H	2	CMWVAR
			BYTE EXTERNAL(59) 340 341 391 392
			BYTE ARRAY(80) EXTERNAL(60) 178 312 319
			340 345 350 364 372 391 399
			LITERALLY 18 19 25 28 31 34 37
			40 43 46 49 52 55 58 64 82 85
			88 91 143 147 149 215
			PROCEDURE BYTE EXTERNAL(20) STACK=0000H
			BYTE 383
			PROCEDURE BYTE EXTERNAL(21) STACK=0000H
			WORD PARAMETER 46
			BYTE PARAMETER AUTOMATIC 203 211
			BYTE MEMBER(OPERANDS)
			LITERALLY
			LITERALLY
			LITERALLY
			LITERALLY
			LITERALLY 189
			BYTE PARAMETER 70
			BYTE PARAMETER 73
			BYTE PARAMETER 76
			BYTE PARAMETER 64
			BYTE PARAMETER 111
			BYTE PARAMETER 79
			BYTE PARAMETER 114
			BYTE PARAMETER 67
			BYTE PARAMETER 108
			WORD EXTERNAL(51)
			PROCEDURE EXTERNAL(46) STACK=0000H
			PROCEDURE EXTERNAL(45) STACK=0000H
			PROCEDURE EXTERNAL(47) STACK=0000H
			PROCEDURE EXTERNAL(43) STACK=0000H
			PROCEDURE EXTERNAL(48) STACK=0000H
			PROCEDURE STACK=0000H
			BYTE ARRAY(100) 152 241 374 378 381
			BYTE 208 220 385
			BYTE BASED(CMPT) 211 256 271 305 310
			WORD 204 205 216 217 241 384
			BYTE PUBLIC 207 219 261 276 314 321
			347 360 411
			BYTE PUBLIC 386 412
			BYTE ARRAY(80) 230 237 377 380 391
			BYTE 230 237 392
			WORD 148 377
			BYTE BASED(CMPT)
			BYTE BASED(CMPT)
			WORD PUBLIC 148 211 212 223 224 256
			257 271 272 304 305 309 310 378
			WORD BASED(CMPT) 223

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

3		CODE	LITERALLY	
148	005CH	1 CODEMACROFOUND . .	BYTE	
8		CODEMACROHEAD . .	LITERALLY	
17		COLON	LITERALLY	345
17		COMMA	LITERALLY	312 326
54	0000H	COPY	PROCEDURE EXTERNAL(14) STACK=0000H	340 391
2		CR	LITERALLY	
147	0000H	1 CSEGSPEC	BYTE EXTERNAL(54)	
147	0000H	1 CSEGTYPE	BYTE EXTERNAL(57)	
147	0000H	2 CSEGVALUE	WORD EXTERNAL(53)	
12		CSOVER	LITERALLY	
147	0000H	2 CURDSEG	WORD EXTERNAL(56)	
87	0000H	2 D	WORD PARAMETER	83
81	0000H	2 D	WORD PARAMETER	82
54	0000H	2 D	WORD PARAMETER	55
84	0000H	2 D	WORD PARAMETER	85
90	0000H	2 D	WORD PARAMETER	91
2		DCL	LITERALLY	
90	0000H	DECOUT	PROCEDURE EXTERNAL(26) STACK=0000H	
3		DELETEDSYM	LITERALLY	
147	0001H	1 DESCR	BYTE MEMBER(TOKEN)	189 191 198
33	0000H	2 DEST	WORD PARAMETER	34
142	0000H	1 DEV	BYTE PARAMETER	143
66	0000H	DIGIT	PROCEDURE BYTE EXTERNAL(18) STACK=0000H	
185	011AH	1 DISP	BYTE	188 190 192 193
3		DOUBLEDEFINED . .	LITERALLY	
16		DOUBLFRANGE . .	LITERALLY	310
308	02DCH	53 DRANGE	PROCEDURE STACK=0014H	327
12		DSOVER	LITERALLY	
147	0000H	1 DSPEC	BYTE EXTERNAL(55)	
4		DWRD	LITERALLY	191 198
2		ELSEDD	LITERALLY	
105	0000H	EMPTYLINE	PROCEDURE BYTE EXTERNAL(32) STACK=0000H	
2		ENDCASE	LITERALLY	
2		ENDDD	LITERALLY	
2		ENDFOREVER	LITERALLY	
2		ENDIF	LITERALLY	179 238
2		ENMODULE	LITERALLY	
2		ENDOFFILE	LITERALLY	
19	0000H	2 ENDOFSYMTAB	WORD EXTERNAL(1)	
2		ENDPROC	LITERALLY	323 414
2		ENDWHILE	LITERALLY	
36	0000H	ENTERATTRIBUTES . .	PROCEDURE EXTERNAL(8) STACK=0000H	
81	0000H	EQUAL	PROCEDURE BYTE EXTERNAL(23) STACK=0000H	153
3		ERROR	LITERALLY	
12		ESOVER	LITERALLY	
2		FALSE	LITERALLY	159 238 242 385 386 390
60	0000H	2 FCBADR	WORD PARAMETER	61
60	0000H	FILEABORT	PROCEDURE EXTERNAL(16) STACK=0000H	
8		FILEISTRUCTURE . .	LITERALLY	
8		FILEOSTRUCTURE . .	LITERALLY	
145	0000H	FILESETUP	PROCEDURE BYTE EXTERNAL(50) STACK=0000H	
63	0000H	FILL	PROCEDURE EXTERNAL(17) STACK=0000H	380 381
				382
142	0000H	2 FILNAM	WORD PARAMETER	143
142	0000H	2 FILTYP	WORD PARAMETER	143
39	0000H	FINDCODEMACRO . . .	PROCEDURE BYTE EXTERNAL(9) STACK=0000H	230

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

30	0000H		FINDSYMBOL	PROCEDURE BYTE EXTERNAL(6) STACK=0000H				
17			FIRST	LITERALLY	254	263	306	311
2			FOREVER	LITERALLY				
396	0479H	29	FORMALLTSTROUT	PROCEDURE PUBLIC STACK=00020H				
2			FORMFEED	LITERALLY				
18	0000H	2	FREET	WORD EXTERNAL(0)		240		
33	0000H		GETATTRIBUTES	PROCEDURE EXTERNAL(7) STACK=0000H				
148	0113H	1	GLOBALCHERROR	BYTE PUBLIC		405		
84	0000H		HEX1OUT	PROCEDURE EXTERNAL(24) STACK=0000H				
87	0000H		HEX2OUT	PROCEDURE EXTERNAL(25) STACK=0000H				
150	0117H	1	I	BYTE	151	153	155	157
171	0119H	1	I	BYTE	173	174	175	176 182
162	0118H	1	I	BYTE	164	165	166	168
3			IDENT	LITERALLY		389	397	
142	0000H		IFILESETUP	PROCEDURE EXTERNAL(49) STACK=0000H				
8			INDEXREGBIT	LITERALLY				
8			INDEXRLGCOUNT	LITERALLY				
118	0000H		ININCLUDEBYTE	PROCEDURE BYTE EXTERNAL(37) STACK=0000H				
2			INIT	LITERALLY				
376	0400H	79	INITCMROUT	PROCEDURE PUBLIC STACK=000AH				
116	0000H		INSOURCEBYTE	PROCEDURE BYTE EXTERNAL(36) STACK=0000H				
8			IREGBIT	LITERALLY				
8			IREGCOUNT	LITERALLY				
170	0004H	1	L	BYTE PARAMETER AUTOMATIC		171	174	176
161	0004H	1	L	BYTE PARAMETER AUTOMATIC		162	165	
335	0004H	1	L	BYTE PARAMETER AUTOMATIC		336	337	
331	0004H	1	L	BYTE PARAMETER AUTOMATIC		332	333	
245	0004H	1	L	BYTE PARAMETER AUTOMATIC		246	247	
3			LAB	LITERALLY				
			LAST	BUILTIN		164	173	
7			LEFTBRACKET	LITERALLY				
17			LEFTPAR	LITERALLY		372		
170	009AH	76	LEGALMODLETTER	PROCEDURE BYTE STACK=0004H			364	
149	0000H	113	LEGALPARAMETER	PROCEDURE BYTE PUBLIC STACK=0012H				
184	00E6H	55	LEGALREGISTER	PROCEDURE BYTE STACK=0002H			274	
195	0110H	27	LEGALSEGREG	PROCEDURE BYTE PUBLIC STACK=0002H				
161	0071H	41	LEGALSPECLETTER	PROCEDURE BYTE STACK=0004H			350	
			LENGTH	BUILTIN		380	381	382
147	0000H	2	LENGTH	WORD MEMBER(OPERANDS)				
69	0000H		LETTER	PROCEDURE BYTE EXTERNAL(19) STACK=0000H				
2			LF	LITERALLY				
149	0008H	1	LG	BYTE PARAMETER AUTOMATIC		150	153	
45	0000H	1	LG	BYTE PARAMETER		46		
42	0000H	1	LG	BYTE PARAMETER		43		
39	0000H	1	LG	BYTE PARAMETER		40		
30	0000H	1	LG	BYTE PARAMETER		31		
27	0000H	1	LG	BYTE PARAMETER		28		
24	0000H	1	LG	BYTE PARAMETER		25		
227	0008H	2	LISTPTR	WORD	227	229	232	233 240
2			LIT	LITERALLY		2	3	4 5 6 7 8
					9	10	11	12 13 14 15 16 17
27	0000H	2	MACDEFPT	WORD PARAMETER			28	
9			MDBF	LITERALLY				
9			MDBIT	LITERALLY				
9			MDBN	LITERALLY				
9			MDDF	LITERALLY				
9			MDDW	LITERALLY				

9		DOWN	LITERALLY	
9		MENDBIT.	LITERALLY	
9		MENDM.	LITERALLY	
9		MFORMALBITS.	LITERALLY	
9		MMODRM1.	LITERALLY	
9		MMODRM2.	LITERALLY	
9		MNOSEGFTX.	LITERALLY	
9		MNUMBERBITS.	LITERALLY	
11		MOOB	LITERALLY	365
11		MOOD	LITERALLY	367
15		MODLETTERBIT	LITERALLY	
15		MODLETTERCOUNT	LITERALLY	
335	031EH	13 MODLETTERROUT.	PROCEDURE STACK=000AH	365 366 367 368
			369	
11		MODSB.	LITERALLY	368
11		MODW	LITERALLY	366
9		MRELB.	LITERALLY	
9		MRELW.	LITERALLY	
9		MSEGFIX.	LITERALLY	
90	0000H	2 N.	WORD PARAMETER	91
87	0000H	2 N.	WORD PARAMETER	88
84	0000H	1 N.	BYTE PARAMETER	95
81	0000H	1 N.	BYTE PARAMETER	82
63	0000H	1 N.	BYTE PARAMETER	64
54	0000H	1 N.	BYTE PARAMETER	55
51	0000H	2 N.	WORD PARAMETER	52
388	044FH	42 NAMEROUT	PROCEDURE BYTE PUBLIC STACK=000AH	
3		NEGLECTED.	LITERALLY	
42	0000H	NEWCMBODY.	PROCEDURE BYTE EXTERNAL(10) STACK=0000H	241
45	0000H	NEWCMNAME.	PROCEDURE BYTE EXTERNAL(11) STACK=0000H	237
27	0000H	NEWMACRO	PROCEDURE BYTE EXTERNAL(5) STACK=0000H	
24	0000H	NEWSYMBOL.	PROCEDURE BYTE EXTERNAL(4) STACK=0000H	
227	0000H	2 NEXT	WORD BASED(LISTPTR)	232 233 240
147	0000H	1 NEXTCH	BYTE EXTERNAL(58)	326
4		NIL.	LITERALLY	
150	0000H	1 NO	BYTE BASED(PTR2)	152
11		NOMOD.	LITERALLY	369
147	0000H	1 NOOPER	BYTE EXTERNAL(61)	
8		NOOVERRCOUNT.	LITERALLY	
8		NOOVERRTDEBIT.	LITERALLY	
13		NOPARAND	LITERALLY	
16		NORANGE.	LITERALLY	
3		NUMBER	LITERALLY	252
16		NUMBERRANGE.	LITERALLY	256
7		OAND	LITERALLY	
7		OEQ.	LITERALLY	
147	0006H	2 OFFSET	WORD MEMBER(COPERANDS)	
7		OGE.	LITERALLY	
7		OGT.	LITERALLY	
7		OLAST.	LITERALLY	
7		OLE.	LITERALLY	
7		OLENGTH.	LITERALLY	
7		OLT.	LITERALLY	
7		OMOD	LITERALLY	
7		ONE.	LITERALLY	
7		ONOT	LITERALLY	
7		OFFSET.	LITERALLY	

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

7      . . . . . LITERALLY
124 0000H OPENHEX. . . . . PROCEDURE EXTERNAL(40) STACK=0000H
122 0000H OPENINCLUDE. . . . . PROCEDURE EXTERNAL(39) STACK=0000H
126 0000H OPENPRINT. . . . . PROCEDURE EXTERNAL(41) STACK=0000H
120 0000H OPENSOURCE. . . . . PROCEDURE EXTERNAL(38) STACK=0000H
128 0000H OPENSMBOL. . . . . PROCEDURE EXTERNAL(42) STACK=0000H
147 0000H 36 OPERANDS. . . . . STRUCTURE ARRAY(4) EXTERNAL(52)
8      OPERANDSTRUC. . . . . LITERALLY 147
3      OPERATOR. . . . . LITERALLY
7      OPTR. . . . . LITERALLY
7      OSEG. . . . . LITERALLY
7      OSHL. . . . . LITERALLY
7      OSHORT. . . . . LITERALLY
7      OSHR. . . . . LITERALLY
7      OTYPE. . . . . LITERALLY
107 0000H OUTHEXBYTE. . . . . PROCEDURE EXTERNAL(33) STACK=0000H
110 0000H OUTPRINTBYTE. . . . . PROCEDURE EXTERNAL(34) STACK=0000H
113 0000H OUTSYMBOLBYTE. . . . . PROCEDURE EXTERNAL(35) STACK=0000H
57 0000H OUTTEXT. . . . . PROCEDURE EXTERNAL(15) STACK=0000H
7      OXOR. . . . . LITERALLY
339 032BH 213 PARDESCROUT. . . . . PROCEDURE STACK=001CH 398
148 005EH 80 PARNAME. . . . . BYTE ARRAY(80) 153 155 379 392
6      PCODEMACRO. . . . . LITERALLY
6      PCSEG. . . . . LITERALLY
6      PDB. . . . . LITERALLY
6      POBIT. . . . . LITERALLY
6      PDD. . . . . LITERALLY
6      PDSEG. . . . . LITERALLY
6      PDW. . . . . LITERALLY
6      PEJECT. . . . . LITERALLY
6      PEND. . . . . LITERALLY
6      PENDIF. . . . . LITERALLY
6      PENDM. . . . . LITERALLY
6      PEQU. . . . . LITERALLY
6      PESEG. . . . . LITERALLY
6      PIF. . . . . LITERALLY
6      PIFLIST. . . . . LITERALLY
6      PINCLUDE. . . . . LITERALLY
6      PLIST. . . . . LITERALLY
6      PMODRM. . . . . LITERALLY
148 0002H 2 PMPT. . . . . WORD 148 340 341 342 343 379
6      PNOTFLIST. . . . . LITERALLY
6      PNOLIST. . . . . LITERALLY
6      PNOSEGFIX. . . . . LITERALLY
6      PORG. . . . . LITERALLY
6      PPAGESIZE. . . . . LITERALLY
6      PPAGEWIDTH. . . . . LITERALLY
6      PRB. . . . . LITERALLY
13      PREFIXAND. . . . . LITERALLY
14      PREFIXON. . . . . LITERALLY
6      PRELB. . . . . LITERALLY
6      PRELW. . . . . LITERALLY
2      PROC. . . . . LITERALLY

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

			20	22	24	27	30	33	36
39	42	45	48	51	72	75	84	87	90
93	95	97	99	102	105	107	110	113	116
118	120	122	124	126	128	130	132	134	136
138	140	142	145	149	161	170	184	195	202

			214	226	245	249	250	303	308	331	335	339
			376	388	396	404						
6		PRS.	LITERALLY									
6		PRW.	LITERALLY									
6		PSEGFIX.	LITERALLY									
3		PSEUDO.	LITERALLY									
6		PSIMFORM.	LITERALLY									
6		PSSEG.	LITERALLY									
63	0000H	2 PT.	WORD PARAMETER	64								
6		PTITLE.	LITERALLY									
228	000AH	2 PTR.	WORD	229	230	237						
149	0006H	2 PTR.	WORD PARAMETER AUTOMATIC		150	153						
42	0000H	2 PTR.	WORD PARAMETER	43								
149	0004H	2 PTR2.	WORD PARAMETER AUTOMATIC		150	152						
202	0138H	44 PUTR.	PROCEDURE PUBLIC STACK=0004H		247	265	333					
			337									
250	0210H	168 PUTRANGE.	PROCEDURE STACK=0010H		306	311	318					
214	0164H	47 PUTW.	PROCEDURE PUBLIC STACK=0004H									
5		RAH.	LITERALLY	291								
5		RAL.	LITERALLY	287								
16		RANGEAND.	LITERALLY									
249	0203H	26 RANGEROUT.	PROCEDURE STACK=0018H		373							
15		RANGESPECBIT.	LITERALLY									
15		RANGESPECCOUNT.	LITERALLY									
16		RANGETYPEAND.	LITERALLY									
5		RAX.	LITERALLY	279								
5		RBH.	LITERALLY	294								
5		RBL.	LITERALLY	290								
5		RBP.	LITERALLY	284								
5		RBX.	LITERALLY	282								
5		RCH.	LITERALLY	292								
5		RCL.	LITERALLY	288								
5		RCS.	LITERALLY	296								
5		RCX.	LITERALLY	280								
5		RDH.	LITERALLY	293								
5		RDI.	LITERALLY	286								
5		RDL.	LITERALLY	289								
5		RDS.	LITERALLY	298								
5		RDX.	LITERALLY	281								
2		REENT.	LITERALLY									
3		REG.	LITERALLY	186	196							
16		REGISTERRANGE.	LITERALLY	271								
245	01F6H	13 REGISTERROUT.	PROCEDURE STACK=000AH		279	280	281	282				
			283	284	285	286	287	288	289	290	291	292
			293	294	295	296	297	298				
5		RES.	LITERALLY	295								
39	0000H	2 RESULT.	WORD PARAMETER	40								
30	0000H	2 RESULT.	WORD PARAMETER	31								
24	0000H	2 RESULT.	WORD PARAMETER	25								
45	0000H	2 RETURNPTR.	WORD PARAMETER	46								
132	0000H	REWINDSOURCE.	PROCEDURE EXTERNAL(44) STACK=0000H									
7		RIGHTBRACKET.	LITERALLY									
17		RIGHTPAR.	LITERALLY	319								
5		RSI.	LITERALLY	285								
5		RSP.	LITERALLY	283								
5		RSS.	LITERALLY	297								
81	0000H	2 S.	WORD PARAMETER	82								

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

54	0000H	2	S.	WORD PARAMETER	50			
95	0000H		SCAN	PROCEDURE EXTERNAL(28) STACK=0000H		301	317	
				325 329 344 349 371 393 401				
93	0000H		SCANINIT	PROCEDURE EXTERNAL(27) STACK=0000H				
17			SECOND	LITERALLY	318			
8			SEGMBIT.	LITERALLY				
8			SEGMCOUNT.	LITERALLY				
147	0004H	2	SEGMENT.	WORD MEMBER(COPERANDS)				
8			SEGTYPEBIT	LITERALLY				
8			SEGTYPECJUNT	LITERALLY				
147	0003H	1	SFLAG.	BYTE MEMBER(COPERANDS)				
16			SINGLEKANGE.	LITERALLY	305			
97	0000H		SKIPRESTOFFLINE	PROCEDURE EXTERNAL(29) STACK=0000H				
102	0000H		SKIPUNITL.	PROCEDURE BYTE EXTERNAL(31) STACK=0000H				
36	0000H	2	SOURCE	WORD PARAMETER	37			
2			SPACE.	LITERALLY				
3			SPEC	LITERALLY				
10			SPECA.	LITERALLY	351			
10			SPECC.	LITERALLY	352			
10			SPEC D.	LITERALLY	353			
10			SPECE.	LITERALLY	354			
99	0000H		SPECIALTOKEN	PROCEDURE BYTE EXTERNAL(30) STACK=0000H				
331	0311H	13	SPECLETERROUT	PROCEDURE STACK=000AH		351	352	353 354
				355 356 357 358				
10			SPECH.	LITERALLY	355			
10			SPECR.	LITERALLY	356			
10			SPECS.	LITERALLY	357			
10			SPECX.	LITERALLY	358			
303	02C5H	23	SRANGE	PROCEDURE STACK=0014H		328		
12			SSOVER	LITERALLY				
39	0000H	2	STRADR	WORD PARAMETER	40			
30	0000H	2	STRADR	WORD PARAMETER	31			
27	0000H	2	STRADR	WORD PARAMETER	28			
24	0000H	2	STRADR	WORD PARAMETER	25			
3			STRING	LITERALLY				
2			STRUC.	LITERALLY	147			
147	0002H	1	STYPE.	BYTE MEMBER(COPERANDS)				
36	0000H	2	SYMBADR.	WORD PARAMETER	37			
33	0000H	2	SYMBADR.	WORD PARAMETER	34			
22	0000H		SYMBINIT	PROCEDURE EXTERNAL(3) STACK=0000H				
3			SYMBOL	LITERALLY				
8			SYMBOLHEAD	LITERALLY				
8			SYMBOLSTRUC.	LITERALLY	147			
20	0000H		SYMBTERMINATE.	PROCEDURE EXTERNAL(2) STACK=0000H				
57	0000H	2	T.	WORD PARAMETER	58			
2			TAB.	LITERALLY				
172	0008H	4	TABLE.	BYTE ARRAY(4) DATA	173 174 176			
163	0000H	8	TABLE.	BYTE ARRAY(8) DATA	164 165			
404	0496H	35	TERMINATECHROUT.	PROCEDURE PUBLIC STACK=000EH				
60	0000H	2	TEXTADR.	WORD PARAMETER	61			
2			THENDO	LITERALLY	176 205 217 230 252 254 259			
				269 312 319 345 405 409				
250	0004H	1	TIME	BYTE PARAMETER AUTOMATIC		251	254	269
102	0000H	1	TOK.	BYTE PARAMETER	103			
99	0000H	1	TOK.	BYTE PARAMETER	100			
147	0000H	4	TOKEN.	STRUCTURE EXTERNAL(57)		186 189 191 193		
				196 198 200 252 259 265 389 397				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		TRUE	LITERALLY	154	207	208	219	220	243	251
				276	314	321	347	360	394	411
147	0000H	1 TYPE	BYTE MEMBER(TOKEN)		186	196	252	339	397	
8		TYPEBIT	LITERALLY							
48	0000H	TYPECALC	PROCEDURE BYTE EXTERNAL(12) STACK=0000H							
8		TYPECOUNT	LITERALLY							
3		UDFSYMB	LITERALLY							
226	0193H	99 UPDATECHLISTS	PROCEDURE BYTE STACK=000AH					409		
78	0000H	UPPER	PROCEDURE BYTE EXTERNAL(22) STACK=0000H							
48	0000H	2 VAL	WORD PARAMETER		49					
147	0002H	2 VALUE	WORD MEMBER(TOKEN)		193	200	259	265		
3		VARIABLE	LITERALLY							
214	0004H	2 W	WORD PARAMETER AUTOMATIC		215	223				
4		WRD	LITERALLY							
51	0000H	WRDTEST	PROCEDURE BYTE EXTERNAL(13) STACK=0000H							

MODULE INFORMATION:

CODE AREA SIZE = 04B9H 1209D
 CONSTANT AREA SIZE = 000CH 12D
 VARIABLE AREA SIZE = 011BH 283D
 MAXIMUM STACK SIZE = 0020H 32D
 936 LINES READ
 0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____